Spectrum One Serial Access Bridge (SAB) II

Programmer's Reference Guide

70-13151-01
Software Version 3.01
Document Revision A
September 1994

symbol

Symbol Technologies, Inc.
116 Wilbur Place
Bohemia, N.Y. 11716

About This Manual

The Spectrum One Serial Access Bridge (SAB) II Programmer's Reference Guide contains programming instructions for accessing host files and applications through the SAB on the Spectrum One network. The target audience for this manual is the host programmer. For information regarding SAB operations, refer to the SAB Quick Reference Guide.

Related Documents

For more information relating to the SAB and the Spectrum One network, refer to the following manuals:

Title	Document Control Number
Serial Access Bridge II User's Guide	70-13150-01
Series 3000 System Software Manual	59045-00-94
Series 3000 Programmer's Reference Manual	59045-00-93
Series 3000 Application Programmer's Guide	59045-00-92
Series 3000 STEP Application Programmer's Guide	59932-00-92
Enabler Programmer's Reference Manual	59886-00-91
Serial Access Bridge II Maintenance Manual	41090-01-00

Some documents listed are provided with network hardware. Others may be ordered through a Symbol sales representative. Refer also to the manuals that are included with the network control unit, terminals, and cradle documentation.

Conventions

Keystrokes are indicated with the angle brackets as follows:

ENTER	identifies a key.
ALT+X	identifies a simultaneous key combination.
BKSP, SHIFT, ON	identifies a key sequence.

Typeface conventions used include.

[brackets]	indicates available command line parameters options or in INI files as separators for configuration options.
Italics	indicates first time a new term is used, a book title, information that must be replaced by an actual value, and menu titles.
Screen	indicates monitor or terminal screen dialog. Also indicates user input.

Notes, Cautions and Warnings

This manual uses notes, cautions and warnings for certain conditions or types of information.

Note: Indicates tips or special requirements.

CAUTION

Indicates conditions that can cause equipment damage or data loss.

Warning: *Warnings indicate procedures that are potentially dangerous and should therefore be performed only by Symbol-authorized repair personnel.*

Contents

Chapter 1

Development Overview

Programming host applications with the Symbol Spectrum One *Serial Access Bridge (SAB)* attached to the host requires knowledge of how the SAB and host communicate. The SAB uses *serial packets* to communicate with the host. With the use of *bridge access routines (BARs)*, the Spectrum One network can be administered from the host. BARs also allow file exchanges between the host and radio terminals.

The basic task of the *host programmer* is to create and interpret serial packets exchanged between the host and SAB. The host application programmer should have a basic familiarity with:

- host application programming
- Symbol Series 3000 terminals
- C language, if using Symbol development tools.

Development tools include:

Symbol Terminal Enabler Program (STEP)	radio terminal program which interprets routines from the enablers. The STEP/enabler combination controls terminal input, output and display. STEP also manages scanner and peripherals connected to the terminal. *Refer to the Series 3000 STEP Application Programmer's Guide*.
host dependant enablers	C routine library for interfacing the host with the Spectrum One network. These include higher-level routines that interact with STEP. The SAB can only use the Symbol serial enablers. Refer to the *Enabler Programmer's Reference Manual*.
Spectrum One Application Development Kit (ADK)	includes interface libraries for C and utilities for creating hex images, mapping the keyboard, and scanning. The kit includes the *Series 3000 Programmer's Reference Manual* and the *Series 3000 Application Programmer's Guide*.

To develop terminal applications without Symbol tools, the host programmer can use information found in the Series 3000 System Software Manual. The manual is a low-level software reference for programming radio terminals including information on the terminal basic input/output system (BIOS), the terminal operating system, printer driver and scanner driver.

SAB Interface and Application Development

The SAB interface controls Spectrum One network operations, host file access services, and communication between host-based applications and terminals. Major SAB interface components include:

Symbol Host Interface Program (SHIP)	Permits communication between a terminal and a host application through the RS-232-C serial link.
BARs	An optional set of a standardized program interface routines accessible from the host. BARs control administrative Spectrum One network operations and provide access to file resources and management on the host for the SAB. BARs are packets containing unique user addresses to communicate with the SAB.

To support the radio network, the SAB runs several distinct software components. The *SAB Server Program (SSP)* organizes network topology and facilitates terminal hand-off from one transceiver to another. SSP packetizes data for error-free delivery to and from the terminals and interfaces with the cradle and transceiver.

Before programming in the SAB environment you should consider some unique features of the Spectrum One SAB.

Sessions and Datagrams

Serial transactions in synchronous mode translate to the NetBIOS datagram mode. Asynchronous mode translates to the NetBIOS session mode. The terminal determines the operative mode for the Spectrum One network

Session requests are more secure and datagram requests are usually more efficient. Session mode provides a delivery guarantee or an error within a specific time interval through NetBIOS support for automatic acknowledgment. A host application should use session requests for reliable data transmission and the minimal data loss.

Connection Establishment and Termination

A terminal to a host process connection occurs when the SAB initiates a Connect Indication. The host responds with a Connect Request. The SAB sends a Set Parameters Indication and the host responds with a Set Parameters Request. The host application is able to exchange data with the terminal until the connection terminates.

To provide network connectivity, a connection is established between the host BAR and the SAB BAR for sending and receiving data and file commands. Once the BAR is received by the SAB or prior to it being passed to the Serial Protocol Driver, it is handled by the SAB software NetBIOS portion.

The connection terminates when the SAB sends a Disconnect Indication and the host responds with a Disconnect Request. The host cannot initiate a Disconnect Request. It would be ignored by the SAB. The host can, however, initiate a Hang Up Request that asks the SAB to initiate a Disconnect Indication.

For host administration using the serial packet protocol, an indication is always sent by the SAB. The host always sends a request even though the SAB initiates the transaction.

Regardless which initiates connection termination, the SAB receives the Disconnect Request and SAB interface terminates communication with the specified terminal.

Data Buffers

When communicating with the host application, SAB interface supports a 512-byte data buffer in datagram mode or a 1024-byte data buffer in session mode. If the data buffer limit is exceeded, SAB interface generates the NetBIOS error code 0x01: Illegal Buffer Length.

Data Format

All short and long integer formats follow the Intel convention. The *least significant bit (LSB)* is positioned in byte 0 and is recieved first. The *most significant bit (MSB)* is in byte 1 for a short integer and in byte three for a long integer.

Command and Response Packet Structure

Many packets are strictly for serial link administration which is not passed to the SAB interface or the terminal. The 256-byte packets includes a header, data and a terminating character.

The header contains addressing and the ucntr field that identifies the particular packet function. Data passes between the host and terminal using specific application data packets. These packets contain 512 bytes in a 2-packet combination. In multi-packet transactions, packets may contain more than 512 bytes.

BAR Enabling

To use BARs on the SAB, Under [BARs] in SSP.INI, set the Admin field to yes to enable administrative BARs, and set Filesrv to yes to enable file access BARs.

BAR Packet Structure

The 8-byte header contains a flag to identify the packet type, a code to define the command or response, the packet data length, and a command specific host field.

Data following the header varies in length up to 512 bytes. Data is command/response specific and no space is allocated for it within the header. Data is part of a command or response packet. For example, a read file command has no data field. The read file command response fills the data field with the file read.

Several administrative BARs are exceptions to the data length of 512 bytes. They still are only able to contain up to 512 bytes in a single packet, but send data in successive packets using a flag bit in byte 0 to indicate that another packet is due to follow.

Information passed between the host and the SAB for BARs are in specific packets identified by ucntr of 70, 71, and 72. A user address (uadr) of 126 identifies the packets as administrative BARs while a uadr of 127 identifies file access BARs.

Host Considerations

The SAB administrative and file access BARs do not link directly with any host application. They are limited to specific serial user addresses. One address is dedicated to support administrative SAB control from the host. Another address supports the host file access BARs that provide file services to the SAB. The SAB views these host components as separate applications.

In addition, each host application that interfaces with the SAB interface is also separate. This is true even if each application maintains individual connections with multiple terminals. A multi-tasking operating system views each application as a separate logical entity and treats it as an individual task.

Host applications using the SAB interface require no special programming efforts in a multi-tasking environment.

SAB interface components behave as follows:

- NetBIOS WAIT mode serializes each request to the host file access BAR from the SAB through the SAB file access BAR. The SAB file access BAR stops while a request is pending and the host file access BAR is not required to field multiple asynchronous requests.
- Administrative BARs process and respond to one command at a time.
- SAB interface takes multiple asynchronous requests from the host. Design host applications that use the SAB interface to manage terminal communication as the individual application dictates.

Host Application Interface

The SAB interface enables communication between a host application on the serial link and terminals on the Spectrum One network with the NetBIOS API packet protocol. A host application uses the serial packet protocol to communicate with the terminals. The SAB interface provides multiplexed access to terminals for the host. Each connection with a terminal is a logical serial packet protocol connection established by each terminal.

Naming Conventions

To communicate with a host process or a terminal, specify the host or remote network address.

The SAB assigns unique terminal names when the terminal is initialized in its cradle. Host names are pre-assigned during SAB initialization. Options under [NetBIOS] in SSP.INI contains one record for each application when the SAB interface is operational.

Host Application Naming

A terminal accesses a host application by its host ID. All host applications have unique names to distinguish their traffic from other computer or process traffic in the network.

The host application name format is:

```
HOSThhhppp
```

where:

- hhh — a unique 3-digit ASCII number from 000 to 999 identifying the host. The SAB supports one host.
- ppp — a unique 3-digit ASCII number from 000 to 999 identifying the host application.

The 16-character name is padded on the right with spaces.

Terminal Naming

A host process contacts a terminal by its terminal name. All terminals have unique names to distinguish their traffic from other terminal traffic in the network. The SAB assigns a terminal address during a terminal log on by sending a Connect Indication packet to the host.

Terminal names are in the following format:

```
SYMBOLrrr
```

where:

rrr a unique 3-digit ASCII number from 000 to 251 identifying the terminal and corresponding to the number assigned by the SAB at initialization.

The 16-character name is padded on the right with spaces.

Naming Conversion

Serial user addresses use 3-digit numeric fields with numbers from 002 through 127. NetBIOS network names use16-character alphanumeric fields. The conversion between them takes place in the serial protocol layer. The serial packet protocol converts the following:

- terminal names
- host process names
- BAR names.

Terminal Names

The serial protocol converts the terminal NetBIOS name *SYMBOLrrr* to a serial terminal ID field *nnnnnn* by dropping *SYMBOL* and padding the field with leading zero characters. *SYMBOLrrr* becomes *000rrr*.

The NetBIOS to serial conversion works in both directions:

- NetBIOS `SYMBOL001` to serial ID `000001`
- Serial ID `000025` to NetBIOS `SYMBOL025`.

The terminal ID is allocated to the next available uadr in an incremental, modulo 64, lowest-available number. The values 000, 126, and 127 are never assigned as a terminal ID. Although the serial terminal ID field is 6 characters, the actual terminal ID is only 3 characters. The terminal ID field is always right-justified and zero filled.

Host Process Names

The host process names are passed through without modification in the Connect Indication transaction. Options under [NetBIOS] in SSP.INI associate the serial uadr with the terminal ID and host process name. SSP.INI routes data between a terminal and a host process.

BAR Names

The serial packet protocol converts the various BAR names in both directions.

File access BARs:

- NetBIOS `SYMHOSTSERVICE0` converts to serial ID `000127`
- Serial ID `000127` converts to NetBIOS `SYMBRIDGSERVICE0`

Administrative BARs:

- NetBIOS `SYMHOSTADMIN0` is converted to serial ID `000126`
- Serial ID `000126` is converted to NetBIOS `SYMBRIDGEADMIN0`.

Network Name Assignment

The network name is derived from the unique terminal unit number. The SAB assigns the unit number and network name that are initially unknown to the host. The SAB interface maintains a remote table for the assignment process and as terminals attach and detach from the network.

SAB initialization defines individual logical connections supported by the serial interface. Host processes do not need to notify the SAB interface of their existence. Terminals must handle all assignments to the host process in the system.

Chapter 2

Serial Packet Protocol Transactions

The serial packet protocol performs all serial I/O routines and all receive and transmit data buffering. NetBIOS internally routes data between the packet protocol and the SAB interface. The SAB usually initiates transactions, but the protocol defines the host as sending a request and the SAB as sending an indication.

The transaction types and the field functions depend on the transaction directions. The transaction types and packet field definitions are organized:

- Packet Field Definitions from SAB to host
- Transactions from SAB to host
- Packet Field Definitions from host to SAB
- Transactions from host to SAB.

Note: Each packet description includes the ucntr number in parenthesis.

A uadr is a number from 001 to 064 indicating the terminal address. Administrative BARs have a uadr of 126. File Access BARs have a uadr of 127. In the table, 0nn indicates the terminal address.

Table 2-1. SAB to Host Packets

uadr	ucntr	Packet
0nn	00	Application data from terminal to host. First packet of 2-packet transaction.
0nn	01	Application data form terminal to host. Last packet of 1 or 2-packet transaction.
0nn	02	Connect Indication.
0nn	03	Disconnect Indication.
0nn	04	Set Parameters Indication.
0nn	05	Positive Acknowledgment Indication.
0nn	06	Negative Acknowledgment Indication.
0nn	60	Application data from terminal to host. First packet in a multi-packet transaction.
0nn	61	Application data from terminal to host. Intermediate packet in a multi-packet transaction.
0nn	62	Application data from terminal to host. Last packet in a multi-packet transaction.
126	70	Administrative BAR Indication. First packet in a multi-packet transaction.
126	71	Administrative BAR Indication. Intermediate packet in a multi-packet transaction.
126	72	Administrative BAR Indication. Last packet in a multi-packet transaction.
127	70	File Access BAR Indication. First packet in a multi-packet transaction.
127	71	File Access BAR Indication. Intermediate packet in a multi-packet transaction.
127	72	File Access BAR Indication. Last packet in a multi-packet transaction.

Table 2-2. Host to SAB Packets

uadr	ucntr	Packet
0nn	00	Application data from host to terminal. First packet of 2-packet transaction.
0nn	01	Application data from host to terminal. Last packet of 1 or 2-packet transaction.
0nn	02	Connect Request.
0nn	03	Disconnect Request.
0nn	04	Set Parameters Request.
0nn	07	Unsolicited Application data From host. First packet of a 2-packet transaction.
0nn	08	Unsolicited Application data From host. Last packet of a 1 or 2-packet transaction.
000	20	Reset Request.
000	21	Hang Up Request.
0nn	60	Application data from host to terminal. First packet in a multi-packet transaction.
0nn	61	Application data from host to terminal. Intermediate packet in a multi-packet transaction.
0nn	62	Application data from host to terminal. Last packet in a multi-packet transaction.
126	70	Administrative BAR Request. First packet in a multi-packet transaction.
126	71	Administrative BAR Request. Intermediate packet in a multi-packet transaction.
126	72	Administrative BAR Request. Last packet in a multi-packet transaction.
127	70	File Access BAR Request. First packet in a multi-packet transaction.
127	71	File Access BAR Request. Intermediate packet in a multi-packet transaction.
127	72	File Access BAR Request. Last packet in a multi-packet transaction.

Packet Field Definitions From SAB To Host

Information is transferred between the SAB and host through the serial packet protocol. A packet is a series of pre-defined data fields containing codes recognized by the SAB and host.

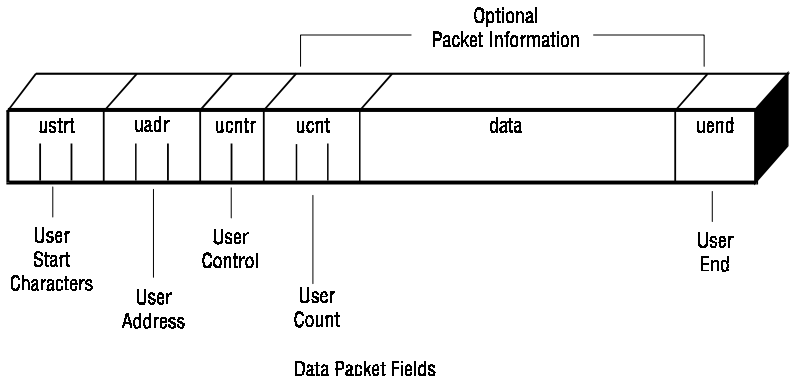


Table 2-3. Packet Description From SAB to Host

Field	Description
ustrt	User start contains ASCII characters STX to indicate the start of the packet.
uadr	User address. A 3-byte field containing the terminal address. The SAB assigns an incremental, modulo 64, lowest-available number. This field can specify up to 64 terminals. Addresses 000, 126 and 127 are reserved for system transactions. The host saves this address for future communication with the terminal assigned to a particular terminal ID. The address is received by the host when the SAB sends a Connect Indication. During terminal log off sequence, the host must send a Disconnect Request (in response to a Disconnect Indication) to allow another terminal to use that particular address.
ucntr	User control. A 2-byte ASCII character field that defines the transaction type. For transactions for the SAB to the host, the values in the ucntr field are as follows:

Table 2-3. Packet Description From SAB to Host

Field	Description	
ucntr	00	data field contains terminal application data. It is the first packet in a 2-packet transaction. The ucnt field contains the number of data field characters.
	01	data field contains terminal application data. It is the only packet in a 1-packet transaction or the second packet in a 2-packet transaction. The ucnt field contains the number of data field characters.
	02	data field contains terminal ID and host ID as part of the Connect Indication Packet. The ucnt contains ASCII value 022 indicating data field character count.
	03	No packet information. This is a Disconnect Indication transaction.
	04	data field contains an ASCII value of 0 or 1 as part of the Set Parameters Indication packet. A value of 0 indicates datagram mode specified by the terminal. A value of 1 indicates session mode. The ucnt field contains the ASCII value of 001.
	05	No packet information. Positive Acknowledgment Indication.
	06	No packet information. Negative Acknowledgment Indication.
	60	data field contains terminal application data. First packet in a multi-packet transaction. The ucnt field contains the number of data field characters.
	61	data field contains terminal application data. It is an intermediate packet in a multi-packet transaction. There can be more than one 61-packets in a transaction. The ucnt field contains the number of data field characters.
	62	data field contains terminal application data. It is the last packet in a multi-packet transaction. The ucnt field contains the number of data field characters.

Table 2-3. Packet Description From SAB to Host

Field	Description	
ucntr	70	data field contains the SAB BAR data. It is the first packet in a multi-packet transaction. Only one 70-packet is allowed in a transaction. The ucnt field contains the number of data field characters. The BAR packets contain binary data and should only be used if the serial link is set to 8-bit data and no parity.
	71	data field contains the SAB BAR data. It is an intermediate packet in a multi-packet transaction. There can be more than one 71-packets in a transaction. The ucnt contains the number of characters in data field.
	72	data field contains the SAB BAR data. It is the last packet in a multi-packet transaction. Only one 72-packet is allowed in a transaction. The ucnt contains the number of data field characters.
Packet Information	The two-field packet information (ucnt and data) contains the actual application data transferred between the terminal and host.	
ucnt	A 3-byte field that contains the data field character count.	
data	This optional application data can be any ASCII characters. Binary data can also be the data field but the application must use the value in the ucnt field to locate the end of the packet instead of searching the Return character in the uend field. The data type in the packet information depends on ucntr.	
uend	User end contains the ASCII Return character (0x0D) to indicate end of packet.	

Transactions From SAB to Host

This section covers the transactions that the terminal can send to the host through the SAB. Some transactions listed are only exchanged between the host and the SAB and may not be seen by the terminal.

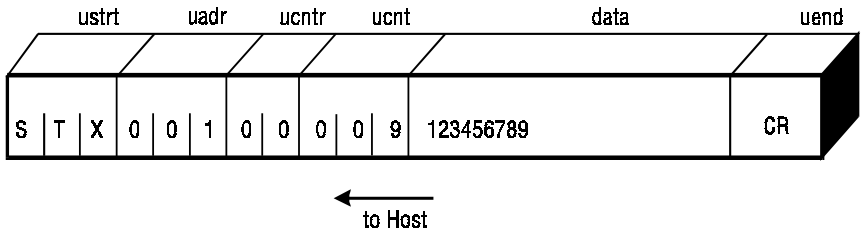
Application Data From Terminal to Host (00, 01)

ucntr: 00 - First packet from terminal to host.

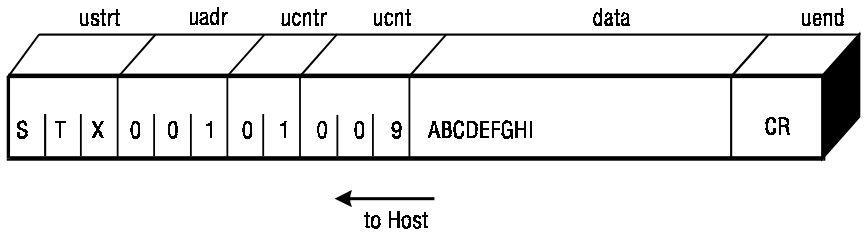
01 - Last packet from terminal to host.

This transaction transmits application data from the terminal to the host via the SAB. The transaction transmits up to two packets. Each packet contains up to 256 bytes of application data in its packet information field.

The packets are used in datagram or session mode.



First Packet with Data from Terminal



Last Packet with Data from Terminal

Connect Indication (02)

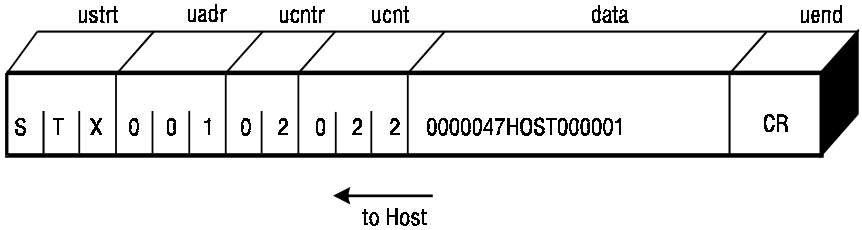
ucntr: 02

Connect Indication allows the SAB to log a terminal to the host. It indicates to the host that a logical connection has been made with the terminal identified in the packet information field.

Terminal ID and host ID log-on information is sent in the packet information field to the host. Terminals that log on to the network make a connection with the host using a Connect Indication Packet.

The Connect Indication starts the datagram or session mode.

The ucnt field is always 022, the total character count of 6-character terminal ID and 16-character Host ID.



Connect Indication Packet

Example:

The log-on information is:

Terminal ID is 000064
Host ID is HOST000001.

The host must save the allocated value of 001 for future communication with this terminal. The host must associate the uadr of 001 with the terminal ID of 064. If more than one process is running on this host, the request process name must be saved for future data routing from the terminal.

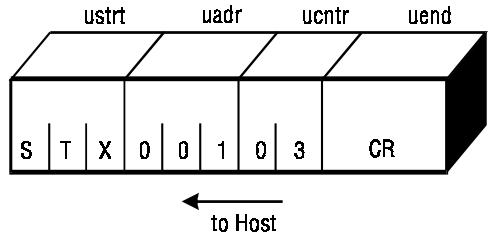
Disconnect Indication (03)

ucntr: 03

The Disconnect Indication allows the SAB to log off a terminal from the host. It indicates to the host that the logical connection has been terminated with the terminal.

The host cannot initiate a disconnect sequence by sending a Disconnect Request packet. The SAB ignores all Disconnect Request packets that are not in response to a Disconnect Indication. However, the host can request that the SAB start a disconnect sequence by sending a Hang Up Request to the SAB.

There is no packet information included in a Disconnect Indication packet. The Disconnect Indication is in datagram or session mode.



Disconnect Indication Packet

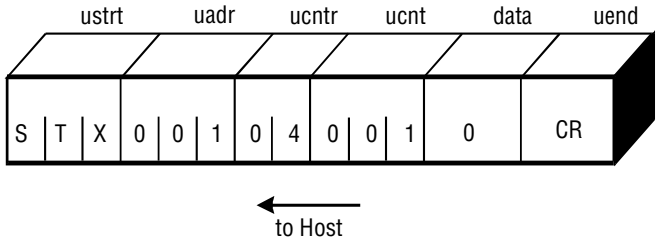
Set Parameters Indication (04)

ucntr: 04

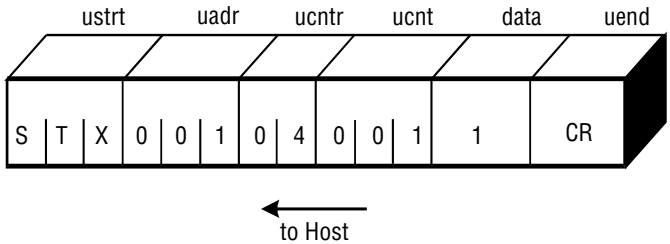
The Set Parameters Indication notifies the host the transmission mode set by the terminal application. Only the terminal can set the transmission mode.

The SAB automatically sends the Set Parameters Indication after receiving the Connect Request from the host. The terminal does not send the Set Parameter Indication over the radio link.

The Set Parameters Indication packet contains a data field value of 0 for datagram mode and 1 for session mode. For this packet, the ucnt field always contains 001.



Set Parameters Indication in Datagram Mode



Set Parameters Indication in Session Mode

Positive Acknowledgment Indication (05)

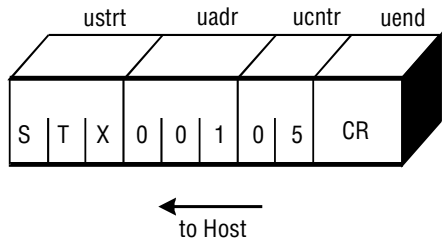
ucntr: 05

The SAB sends a Positive Acknowledgment Indication to the host when it receives one of the following:

- unsolicited host transaction in datagram mode.
- host transaction in session mode.
- Call Sign Request.
- Hang Up Request
- Reset Request.

The acknowledgment does not indicate that the terminal received the data. It only indicates the correct data was received and passed on to the Spectrum One network.

The Positive Acknowledgment Indication packet does not contain any optional packet information.



Positive Acknowledgment Indication

Negative Acknowledgment Indication (06)

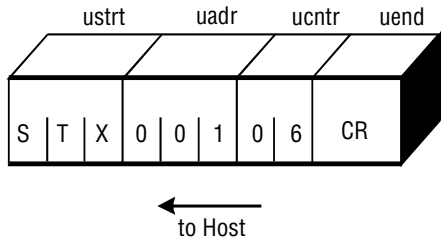
ucntr: 06

The SAB sends a Negative Acknowledgment Indication to notify the host that the terminal failed to receive one of the following:

- unsolicited host transaction in datagram mode.
- host transaction in session mode.
- Hang Up Request
- Reset Request.

It is not a Spectrum One network trouble indicator. There is no direct indication of delivery problems on the Spectrum One network side.

The Negative Acknowledgment Indication packet does not contain any optional packet information.



Negative Acknowledgment Indication

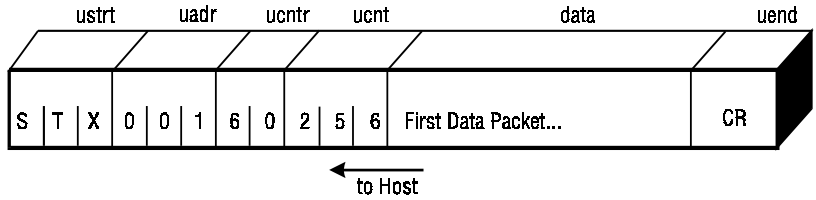
Application Data Greater Than 512-Bytes from Terminal (60, 61, 62)

ucntr: 60 - First packet from terminal to host.

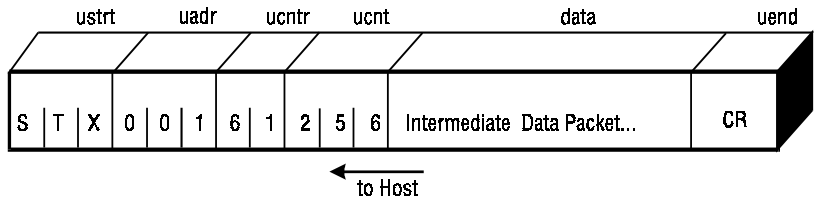
61 - Intermediate packet from terminal to host.

62 - Last packet from terminal to host.

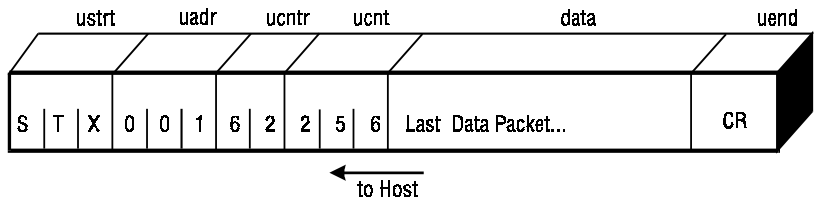
This transaction transmits application data from the terminal to the host when the data exceeds 512 bytes. Up to four packets can be transmitted in one transaction (1024 bytes total). These packets are only allowed in asynchronous mode since datagrams are limited to a maximum of 512 bytes.



First Packet with Data from Terminal



Intermediate Packet with Data from Terminal



Last Packet with Data from Terminal

Administrative BAR Response Indication (70, 71, 72)

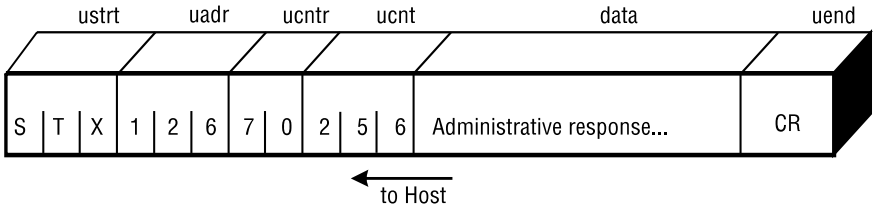
uadr: 126

ucntr: 70 - First packet of Administrative Bar Response

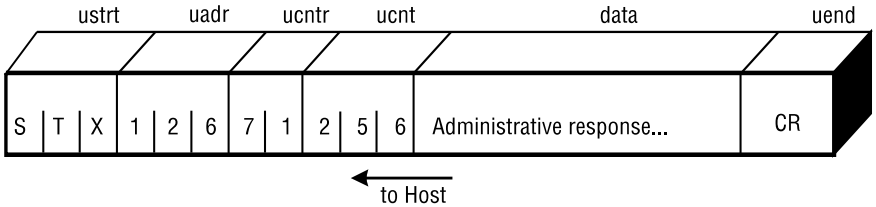
71 - Intermediate packet of Administrative Bar Response

72 - Last packet of Administrative Bar Response

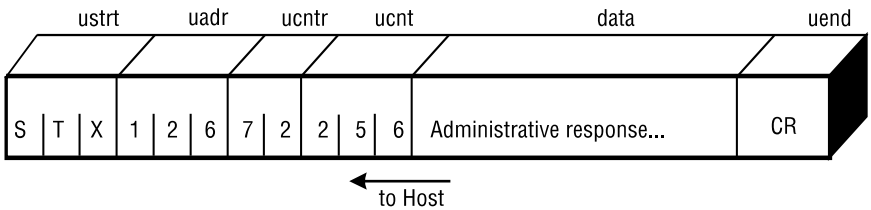
The Administrative BAR Response Indication transfers BAR data between the SAB and host. These packets use uadr 126 (BAR Administrative Service) and uadr 127 (BAR File Service). Both operate in session mode. BAR packets contain binary data and should only be used if the link is set to 8 data bits and no parity.



Administrative BAR Response Indication, First Packet



Administrative BAR Response Indication, Intermediate Packet



Administrative BAR Response Indication, Last Packet

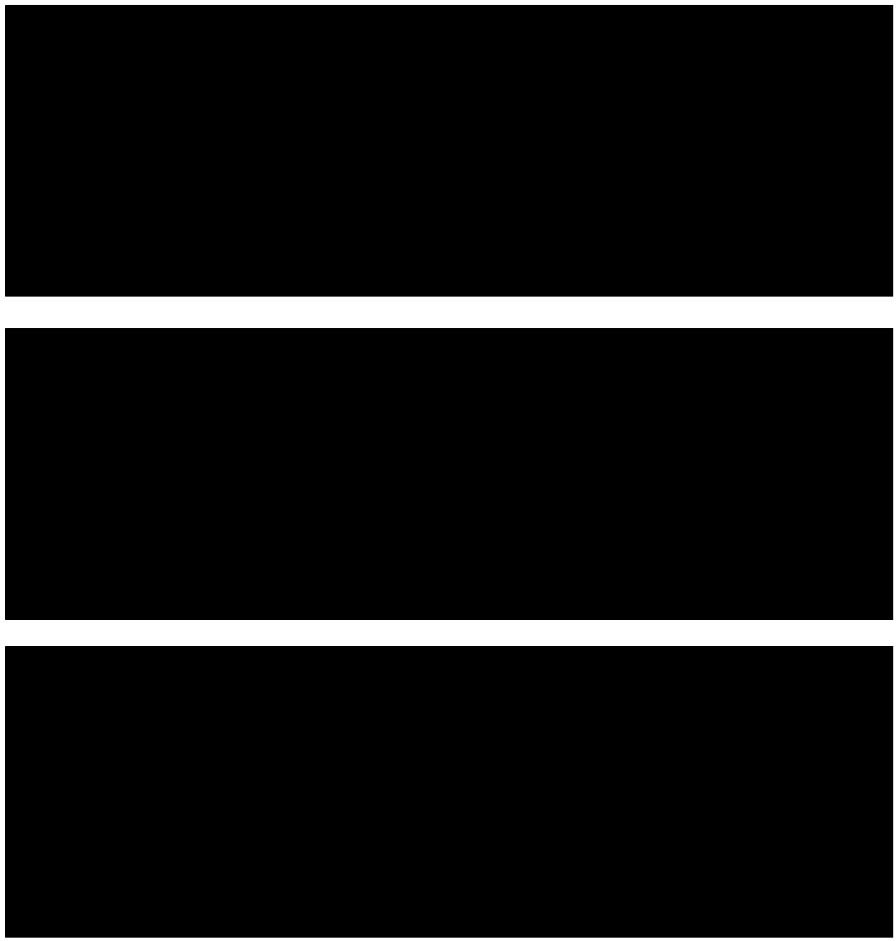
File Service BAR Command Indication (70, 71, 72)

uadr: 127

ucntr: 70 - First packet of File Service BAR command.

71 - Intermediate packet of File Service BAR command.

72 - Last packet of File Service BAR command.



Packet Field Definitions From Host to SAB

Information is transferred between the SAB and host through the serial packet protocol. A packet is a series of pre-defined data fields containing codes recognized by the SAB and host.

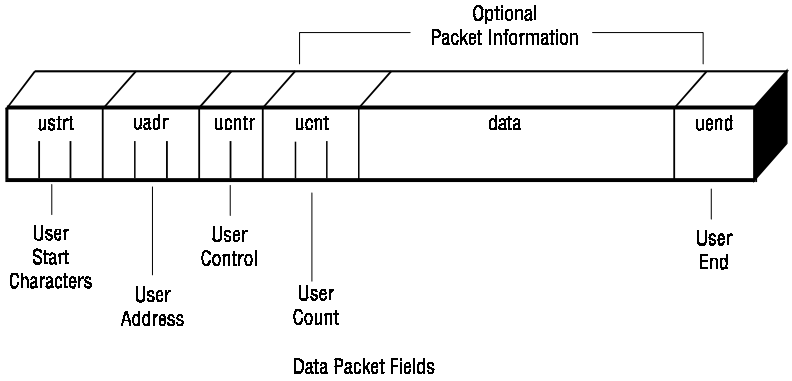


Table 2-4. Packet Description From Host to SAB

Field	Description
ustrt	User start contains ASCII characters STX to indicate the start of the packet.
uadr	User address. A 3-byte field containing the terminal address. The SAB assigns an incremental, modulo 64, lowest-available number. This field can specify up to 64 terminals. Addresses 000, 126 and 127 are reserved for system transactions. The host saves this address for future communication with the terminal assigned to a particular terminal ID. The address is received by the host when the SAB sends a Connect Indication. During terminal log off sequence, the host must send a Disconnect Request (in response to a Disconnect Indication) to allow another terminal to use that particular address.
ucntr	User control. A 2-byte ASCII character field that defines the transaction type. For transactions for the SAB to the host, the values in the ucntr field are as shown below:

Table 2-4. Packet Description From Host to SAB

Field	Description	
ucntr	00	data field contains terminal application data. It is the first packet in a 2-packet transaction. The ucnt field contains the number of data field characters.
	01	data field contains terminal application data. It is the only packet in a 1-packet transaction or the second packet in a 2-packet transaction. The ucnt field contains the number of data field characters.
	02	data field contains terminal ID and host ID as part of the Connect Indication Packet. The ucnt field contains the ASCII value 022 indicating the data field character count.
	03	No packet information. This is a Disconnect Indication transaction.
	04	data field contains an ASCII value of 0 or 1 as part of the Set Parameters Indication packet. A value of 0 indicates datagram mode specified by the terminal. A value of 1 indicates session mode. The ucnt field contains the ASCII value of 001.
	07	data field contains unsolicited application data from the host. It is the first packet in a 2-packet transaction. The unct field contains the number of bytes in the data field.
	08	data field contains unsolicited application data from the host. It is the last packet in a 2-packet transaction. The unct field contains the number of bytes in the data field.
	20	No packet information. Reset Request.
	21	data field contains 3-character terminal uadr required for hang up request.
	50	data field contains 3-character call sign identification message.
	60	data field contains host application data. First packet in a multi-packet transaction. The ucnt field contains the number of data field characters.

Table 2-4. Packet Description From Host to SAB

Field	Description	
ucntr	61	data field contains host application data. It is an intermediate packet in a multi-packet transaction. There can be more than one 61-packets in a transaction. The ucnt field contains the number of data field characters.
	62	data field contains host application data. It is the last packet in a multi-packet transaction. The ucnt field contains the number of data field characters.
	70	data field contains the SAB BAR data. It is the first packet in a multi-packet transaction. Only one 70-packet is allowed in a transaction. The ucnt field contains the number of data field characters. The BAR packets contain binary data and should only be used if the serial link is set to 8-bit data and no parity.
	71	data field contains the SAB BAR data. It is an intermediate packet in a multi-packet transaction. There can be more than one 71-packets in a transaction. The ucnt field contains the number of data field characters.
	72	data field contains the SAB BAR data. It is the last packet in a multi-packet transaction. Only one 72-packet is allowed in a transaction. The ucnt field contains the number of data field characters.
Packet Information	The two-field packet information (ucnt and data) contains the actual application data transferred between the terminal and host.	
ucnt	A 3-byte field that contains the data field character count.	
data	Optional data can be any ASCII characters. Binary data is possible but the application must use the value in the ucnt field to locate the end of the packet instead of searching the Return character in the uend field. The data type in the packet information depends on ucntr.	
uend	User end contains the ASCII Return character (0x0D) to indicate end of packet.	

Transactions From Host To SAB

This section covers the transaction types that the host can send to the terminal through the SAB. Some transactions listed are only exchanged between the host and the SAB. The terminal never sees certain link administration transactions.

Application Data From Host to Terminal (00, 01)

ucntr: 00 - First packet from terminal to host.

01 - Last packet from terminal to host.

This transaction transmits application data from the terminal to the host via the SAB. The transaction transmits up to two packets. Each packet contains up to 256 bytes of application data in its packet information field.

The packets are used in datagram or session mode.

Connect Request (02)

ucntr: 02

The Connect Request indicates that it has accepted the log-on information from the SAB. It indicates to the SAB that a logical connection with the terminal identified by the data in the packet information field has been confirmed.

Terminal ID and host ID log-on information is sent in the packet information field to the SAB. The terminal ID and host ID values must be identical to those received by the host from the SAB in the Connect Indication packet. The response is handled by the SAB. It is not sent to the terminal.

The ucnt field is always 022, the total character count of 6-character Terminal ID and 16-character Host ID.

The Connect Indication starts the datagram or session mode.

Example:

The log-on information is:

Terminal ID is 000064
Host ID is HOST000001.

The host must save the allocated value of 001 for future communication with this terminal. The host must associate the uadr of 001 with the terminal ID of 064. If more than one process is running on this host, the request process name must be saved for future data routing from the terminal.

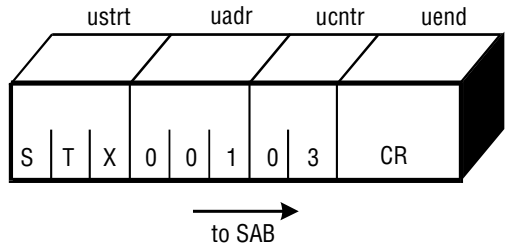
Disconnect Request (03)

ucntr: 03

After the host receives a Disconnect Indication, it sends the Disconnect Request to the SAB to complete the log-off operation initiated by the terminal. The Disconnect Request ends the logical connection between the terminal and host.

The host cannot initiate a disconnect sequence by sending a Disconnect Request packet. The SAB ignores all Disconnect Request packets that are not in response to a Disconnect Indication. However, the host can request that the SAB start a disconnect sequence by sending a Hang Up Request.

There is no packet information (ucnt and data fields) included in a Disconnect Request packet. The Disconnect Request is in datagram or session mode.



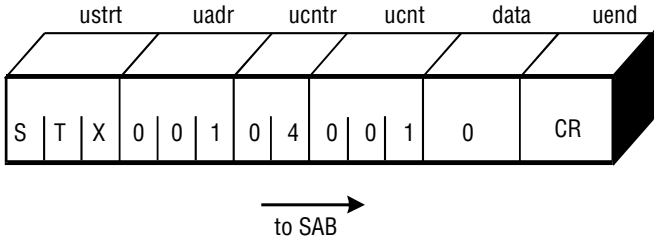
Disconnect Request Packet

Set Parameters Request (04)

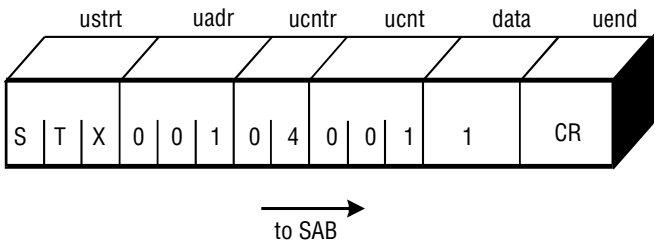
ucntr: 04

The Set Parameters Request is sent by the host after it receives a valid Set Parameters Indication from the SAB. The host sends the Set Parameters Request to confirm the terminal transmission mode selection. Only the terminal can set the transmission mode.

The Set Parameters Request packet contains a data field value of 0 for datagram mode and 1 for session mode. For this packet, the ucnt field always contains 001.



Set Parameters Request in Datagram Mode



Set Parameters Request in Session Mode

Unsolicited Application Data From Host (07, 08)

ucntr: 07 - First Unsolicited packet from host to terminal

08 - Last Unsolicited packet from host to terminal

This transaction in datagram mode allows the host to send unsolicited data through the SAB to the terminal. The unsolicited transaction can not be used in session mode.

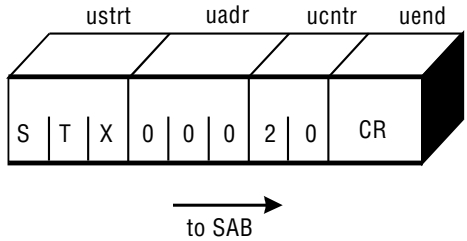
The transaction transmits up to two unsolicited packets. Each packet can contains up to 256 bytes of application data in its packet information field.

Reset Request (20)

ucntr: 20

The host sends a Reset Request when a total link-layer reset is required. The SAB responds with an Acknowledgment Indication and then both the host and the SAB reset all connections. No other transactions are required after the Reset Request and Positive Acknowledgment Indications.

This transaction is an original serial packet protocol extension and is not required for normal operation.



Reset Request Packet

Hang Up Request (21)

ucntr: 21

The host sends a Hang Up Request when a specific terminal requires disconnection. The SAB responds with an Acknowledgment Indication. The SAB sends a Disconnect Indication. A normal disconnect sequence should then be followed by the host.

This transaction is an original serial packet protocol extension and is not required for normal operation.

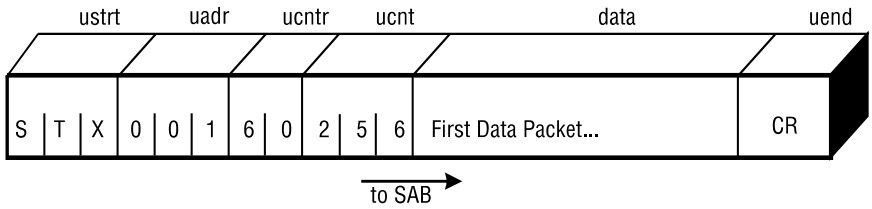
Application Data Greater Than 512-Bytes from Host (60, 61, 62)

ucntr: 60 - First packet from host to terminal.

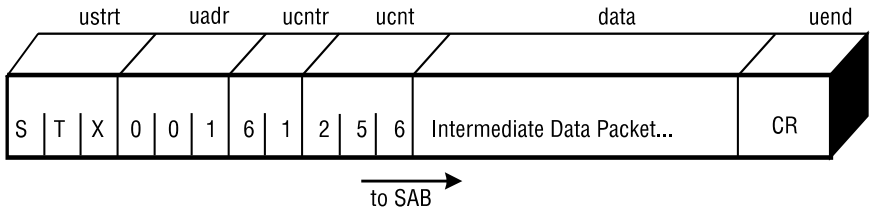
61 - Intermediate packet from host to terminal.

62 - Last packet from host to terminal.

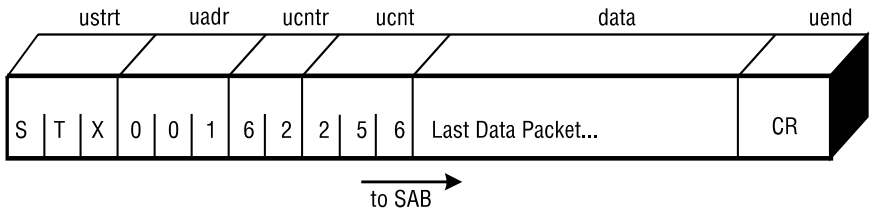
This transaction transmits application data from the host to the terminal when the data exceeds 512 bytes in length. Up to four packets can be transmitted in one transaction (1024 bytes total length). These packets are only allowed in session mode since datagrams are limited to a maximum of 512 bytes.



First Packet with Data From Host



Intermediate Packet with Data From Host



Last Packet with Data From Host

Administrative BAR Command Request (70, 71, 72)

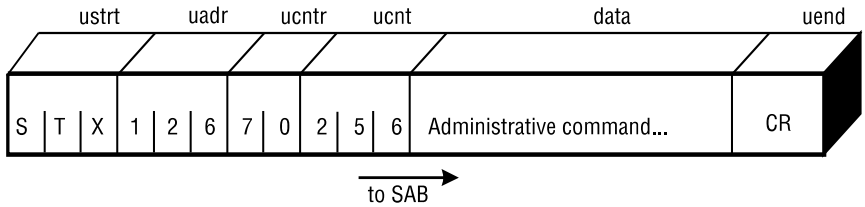
uadr: 126

ucntr: 70 - First packet of Administrative Bar Command

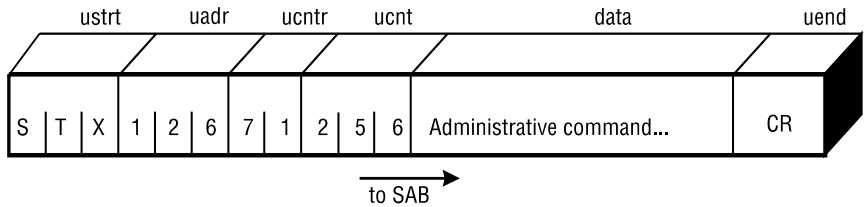
71 - Intermediate packet of Administrative Bar Command

72 - Last packet of Administrative Bar Command

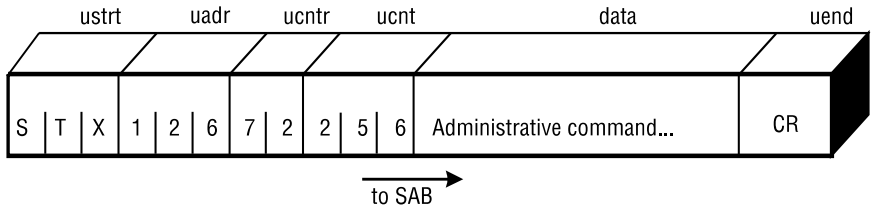
This packet transfers BAR specific data between the host and the SAB. These packets use uadr 126 (BAR Administrative Service) and uadr 127 (BAR File Service), Both operate in session mode. BAR packets contain binary data and should only be used if the link is set to 8 data bits and no parity.



Administrative BAR Command Request, First Packet



Administrative BAR Command Request, Intermediate Packet



Administrative BAR Command Request, Last Packet

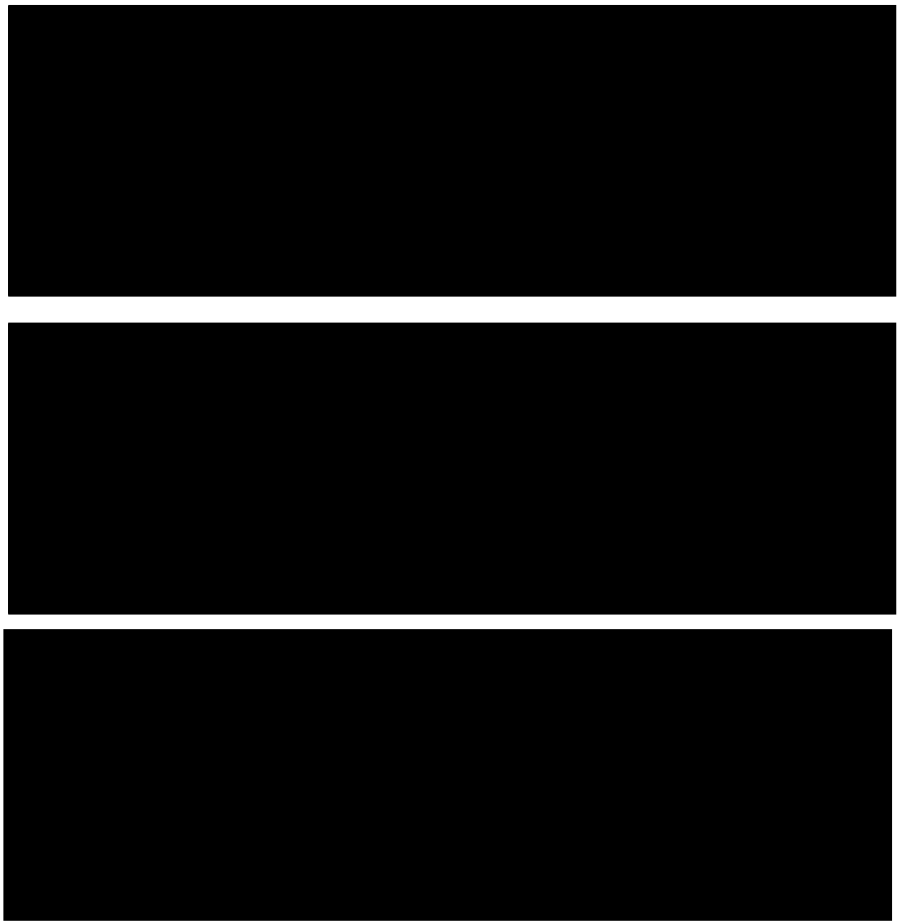
File Service BAR Response Request (70, 71, 72)

uadr: 127

ucntr: 70 - First packet of File Service BAR command.

71 - Intermediate packet of File Service BAR command.

72 - Last packet of File Service BAR command.



Chapter 3

BAR Reference

The BAR reference gives detailed information of BAR commands and their structure within a packet. To use File Access and Administrative BARs, set the directives under [BARs] in SSP.INI to **yes**.

Table 3-1. Command/Response Header Structure

Byte	Field	Contents		
0	Flag Field	Bit 0 identifies command or response packet. 0 = command, 1 = response Bit 1 identifies communication mode: 0 = wait mode, 1 = no-wait mode Other flags are command specific.		
1	Command and Return Code	Individual command or return packet. For administration BARs, command code identifies packet as admin BAR, bytes 4 and 5 identify the actual command. With a response, byte 1 contains an error or status code.		
		0x01	BAR_CREATEFILE	create file
		0x02	BAR_OPENFILE	open file
		0x03	BAR_CLOSEFILE	close file
		0x04	BAR_READDATA	read data from file
		0x05	BAR_WRITEDATA	write data to file
		0x06	BAR_SEEKFILE	seek file
		0x07	BAR_GETLINE	read text line from file
		0x08	BAR_PUTLINE	write text line to file
		0x09	BAR_BRIDGEMSG	send message
		0x0A	BAR_GETTIME	get host time and date
		0x40	BAR_ADMIN	administrative request
2	Data Length LSB	16-bit unsigned integer value giving the data transferred size.		

Table 3-1. Command/Response Header Structure

Byte	Field	Contents
3	Data Length MSB	If no data is sent with the packet, data length equals zero.
4-7	Command Specific	Contains file handle for many file access commands. For administrative BARs, bytes 4 and 5 contain function and sub-function codes while bytes 6 and 7 contain the message length integer value.

File Access BARs

File access BARs interface to and from the host for file services. BARs exist on both the host and the SAB. The host BARs provide location independent file resources and services to the SAB. The BARs on the SAB and the BARs on the host form a client/server relationship. The SAB interface is the client. The host-based BARs services the file access requests. File access BARs provide the following functions:

- create, open and close files
- read and write data from and to files
- seek to a position in a file
- read and write text lines to and from files
- send messages to the host
- obtain date and time from the host.

The SAB processes all file access commands in the client mode. The SAB issues command packets to the host. The host issues response packets when command processing completes.

Create File

Requests host create a named file in the host file system.

The file pathname is not given in the data string. The host provides a default area for the files necessary for SAB operation. If the specified file exists when the Create File command is issued, the file is truncated to 0 length and all data in the existing file is lost.

The Create File command differs from the Open File command. Non-existent files can be created. Existing file are initialized with a length of 0.

No other host-based processes may access the specified file other than SAB requests.

Table 3-2. Create File Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 1 command Packet bit 1 reserved bit 4 = 1 create text mode file = 0 create binary mode file
1	0x01	command code	BAR_CREATFILE
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	DC	unused	

The data field contains the filename to open, represented as an ASCII null-terminated string.

Table 3-3. Create File Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit0 = 0 response packet bit 1 reserved bit 2 = 1 create text mode file = 0 create binary mode file
1	VAR	return code	Refer to Table 3-4.
2	0x00	data length	
3	0x00	data length	
4-7	XXX	file handle	

On subsequent WRITE and CLOSE commands the SAB includes the file handle to indicate to the host which file the command pertains to.

No data field returns with this reply.

Table 3-4. Create File Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x02	BARERR_NOCREFIL	could not create file
0x05	BARERR_OPERATIO N	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Open File

Requests host open a named file in the host file system. The Open File command opens the file for read, with the file pointer at the beginning of the file.

The file pathname is not given in the data string. The host provides a default area for the files necessary for SAB operation.

Table 3-5. Open File Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1 reserved bit 4 = 1 open text mode file = 0 open binary mode file
1	0x02	command code	BAR_OPENFILE
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	unused	

The data field contains the filename to open, represented as an ASCII NULL-terminated string.

Table 3-6. Open File Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 =0 response packet bit 1 reserved bit 4 = 1 open text mode file = 0 open binary mode file
1	VAR	return code	Refer to Table 3-7
2	0x00	data length	
3	0x00	data length	
4-7	XXX	file handle	

On subsequent WRITE and CLOSE commands the SAB includes the file handle to indicate to the host which file the command pertains to.

No data field returns with this reply.

Table 3-7. Open File Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x03	BARERR_FILNOTFND	could not find file
0x05	BARERR_OPERATIO N	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Close File

Requests host close a previously opened file in the host file system. The file handle returned from the OPEN FILE or CREATE FILE commands identifies the file.

Table 3-8. Close File Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 =1 command packet bit 1 reserved
1	0x03	command code	BAR_CLOSEFILE
2	0x00	data length	
3	0x00	data length	
4-7	XXX	file handle	

No data field is attached for this command.

Table 3-9. Close File Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1 reserved bit 4 = 1 create text mode file = 0 create binary mode file
1	0x02	return code	Refer to Table 3-10
2	0x00	data length	
3	0x00	data length	
4-7	XXX	file handle	

Table 3-10. Close File Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x04	BARERR_BADFILHAN	invalid file handle given
0x05	BARERR_OPERATIO N	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Read Binary Data From File

Requests host read a data block from a previously opened file in the host file system. The file handle returned from the OPEN FILE command identifies the file.

The file must be opened in a binary mode with no conversions performed by the host.

Table 3-11. Read Binary Data From File Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 1 command packet bit 1 reserved
1	0x04	command code	BAR_READDATA
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	file handle	

The data field contains the maximum read length for the request. The integer data is represented as an ASCII numeric, NULL-terminated string. Maximum read length is 512 bytes.

Table 3-12. Read Binary Data From File Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 0 response packet bit 1 reserved
1	VAR	return code	Refer to Table 3-13
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	file handle	

The data length header field (bytes 2 and 3) gives the actual length of data read from the host file.

The data field contains the file data read, up to the maximum requested.

Table 3-13. Read Binary Data From File Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x04	BARERR_BADFILHAN	invalid file handle given
0x05	BARERR_OPERATIO N	host operation error occurred
0x05	BARERR_UNKNOWN	unimplemented BAR error

Write Binary Data to File

Requests host write a data block to a previously opened file in the host file system. The file handle returned from the CREATE FILE command identifies the file.

Table 3-14. Write Binary Data to File Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1 reserved
1	0x05	command code	BAR_WRITEDATA
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	file handle	

The file must be opened in a binary mode with no conversions by the host. The data field contains data written to the file. Data length can be 512 bytes maximum.

Table 3-15. Write Binary Data to File Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 0 response packet bit 1 reserved
1	VAR	return code	Refer to Table 3-16
2	0x00	data length	
3	0x00	data length	
4-7	XXX	file handle	

No data field returns with this reply.

Table 3-16. Write Binary Data to File Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x04	BARERR_BADFILHAN	invalid file handle given
0x05	BARERR_OPERATIO N	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Seek Bits

Requests host seek to a random position within a previously opened file in the host file system. The file handle returned from the OPEN FILE or CREATE FILE commands identifies the file.

Table 3-17. Seek Bits Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 1command packet bit 1 reserved bits 5,6 seek start position: 00 = from start of file 01 = from current position 10 = from end of file
1	0x06	command code	BAR_SEEKFILE
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	file handle	

The data field contains the seek offset, from the position specified by flag bits 5 and 6, for this request. This long integer data is represented as an ASCII numeric, NULL-terminated string.

Table 3-18. Seek Bits Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 0 response packet bit 1-3 reserved bit 4 1 = text file, 0 = binary file bits 5,6 seek start position: 00 = from start of file 01 = from current position 10 = from end of file
1	VAR	return code	Refer to Table 3-19
2	0x00	data length	
3	0x00	data length	
4	XXX	file handle	

No data field returns with this reply.

Table 3-19. Seek Bits Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x04	BARERR_BADFILHAN	invalid file handle given
0x05	BARERR_OPERATIO N	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Read Text Line From File

Requests host read a single text line from a host text file.

A text line is defined as ASCII printable data, terminated by a linefeed (LF) and a NULL character. The file handle returned from the OPEN FILE command identifies the file.

The file must have been opened in text mode. The conversion to and from the text line format must be provided on the host.

Table 3-20. Read Text Line From File Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 1 command packet bit 1 reserved
1	0x07	command code	BAR_GETLINE
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	file handle	

The data field contains the maximum read length for this request.

The integer data is represented as an ASCII numeric, NULL-terminated string.

Table 3-21. Read Text Line From File Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1 reserved
1	VAR	return code	Refer to Table 3-22
2	0x00	data length	LSB
3	0x00	data length	MSB
4-7	XXX	file handle	

The data length header field (bytes 2 and 3) gives the actual length of data read from the host file, up to and including the terminating NULL character. The data field contains the file data read, up to the maximum requested. The data is terminated with a linefeed and NULL characters.

Table 3-22. Read Text Line From File Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x04	BARERR_BADFILHAN	invalid file handle given
0x05	BARERR_OPERATIO N	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Write Text Line To File

Requests host write a text line to a previously opened text file in the host file system.

A text line is defined as ASCII printable data, terminated by a linefeed (LF) and Null character. The file handle returned from the CREATE FILE command identifies the file.

The file must have been opened in text mode. The conversion to and from the text line format must be provided on the host.

Table 3-23. Write Text Line To File Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 1 command packet bit 1 reserved
1	0x08	command code	BAR_PUTLINE
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	file handle	

The data field contains the data written to the file. Data length is a maximum of 512 bytes up to and including the terminating NULL character.

Table 3-24. Write Text Line To File Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 0 response packet bit 1 reserved
1	VAR	return code	Refer to Table 3-25
2	0x00	data length	
3	0x00	data length	
4-7	XXX	file handle	

No data field returns with this reply.

Table 3-25. Write Text Line To File Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x04	BARERR_BADFILHAN	invalid file handle given
0x05	BARERR_OPERATIO N	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Send Message

Requests host accept a message in a text stream form. Flags are used by SAB to classify message into one of 16 groups. In general, text string informs host of SAB status or errors.

Table 3-26. Send Message Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 response packet bit 1 reserved bits 4-7 message type 0000 = informational 0001 = SAB fatal error 0011 = log message
1	0x09	command code	BAR_BRIDGEMSG
2	VAR	data length	LSB
3	VAR	data length	MSB
4-7	XXX	unused	

The data field contains the text of the delivered message, represented as a NULL-terminated ASCII string. No response message is generated for this command.

Get Host Date and Time

Requests host return date, time and day, as three ASCII NULL-terminated strings in the response packet data field.

Table 3-27. Get Host Date and Time Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1 reserved
1	0x0A	command code	BAR_GETTIME
2	0x00	data length	
3	0x00	data length	
4-7	XXX	unused	

The data length fields (bytes 2 and 3) are set to a value of zero. No data field is attached to this packet.

Table 3-28. Get Host Date and Time Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1 reserved
1	VAR	return code	Refer to Table 3-29
2	0x00	data length	LSB
3	0x14	data length	MSB
4-7	XXX	unused	

The data length header field (bytes 2 and 3) specifies a length of 20 (0x14), the actual length of the three ASCII, NULL-terminated strings returned for the date, time and day. The date is returned in the first string, the time is returned in the second string and the day is returned in the third, all as ASCII, NULL-terminated strings with the following formats:

mm-dd-yy	mm– represents month (01-12) dd – represents day (01-31) yy – represents year (80-79) (1980-2079)
hh:mm:ss	hh – represents hour (00-23) mm – represents minutes (00-59) ss – represents seconds (00-59)
d	d – represents day (0-6) (0 is Sunday)

Table 3-29. Get Host Date and Time Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x05	BARERR_OPERATION	host operation error occurred
0x06	BARERR_UNKNOWN	unimplemented BAR error

Administrative BARs

The SAB and Spectrum One network require certain administrative functions be performed. The administrative BARs fulfill the SAB remote operational requirements. The administrative BARs exist on both the host and the SAB and provide the following functions:

- standardized interface between the host and the SAB
- network command and control:
  - examine/change radio channels
  - examine/remove terminals
  - examine/remove transceivers
  - restart
- get topology
- get/clear statistics
- get SAB version.

The SAB processes all administrative commands in the server mode. The host issues command packets to the SAB. The SAB issues response packets when command processing completes. When a command is processed a response to the host is always returned.

Get Units

Requests unit numbers as bit-field information for assigned terminals and transceivers.

Table 3-30. Get Units Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0= 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	0x00	data length	
3	0x00	data length	
4	0x00	function code	FUNC_GETINFO
5	0x20	subfunction code	SFUNC_UNITS
6	0x00	message length	LSB
7	0x00	message length	MSB

No data field is sent with this packet.

Table 3-31. Get Units Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-32
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x00	function code	FUNC_GETINFO
5	0x20	subfunction code	SFUNC_UNITS
6	VAR	message length	LSB
7	VAR	message length	MSB

The data field returned with this packet is a bit-field form containing information on currently assigned remote unit numbers. The data is formatted into 512 bits (64 bytes) with each bit representing a transceiver or remote.

Transceivers are assigned numbers (and corresponding bits) in the 1 to 63 range. Remotes are in the 65 to 511 range. A unit number of 0 is assigned to uninitialized transceivers and a unit number is 64 is assigned to uninitialized remotes.

Table 3-32. Get Units Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet

Delete Specific Remote Units

Requests SAB remove assigned remote units specified in the command packet data field.

Table 3-33. Delete Specific Remote Units Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x01	function code	FUNC_SETINFO
5	0x21	subfunction code	SFUNC_DELUNITS
6	VAR	message length	LSB
7	VAR	message length	MSB

The data field contains the all remotes unit numbers selected for removal from the current configuration. The data is formatted as a unsigned-integer variable number (2 bytes each) that represent the unit numbers to delete. The entire packet is rejected if the numbers do not range from 65 to 511.

Table 3-34. Delete Specific Remote Units Response Packet

Byte	Value	Description	Details
0	VAR	flags bit 1-7	bit 0 = 0 response packet reserved
1	VAR	return code	Refer to Table 3-35
2	0x00	data length	
3	0x00	data length	
4	0x01	function code	FUNC_SETINFO
5	0x21	subfunction code	SFUNC_DELUNITS
6	0x00	message length	
7	0x00	message length	

No data field returns with this response.

Table 3-35. Delete Specific Remote Units Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x02	BAREER_BADDATA	invalid unit number

Get Radio Frequencies

Requests SAB return the currently defined radio channels and associated signal strengths. The channels are previously selected by default or using the `SSP.INI` file to define them.

Table 3-36. Get Radio Frequencies Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	0x00	data length	
3	0x00	data length	
4	0x00	function code	FUNC_GETINFO
5	0x02	subfunction code	SFUNC_RADIO
6	0x00	message length	
7	0x00	message length	

No data field is sent with this packet.

Table 3-37. Get Radio Frequencies Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-38
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x00	function code	FUNC_GETINFO
5	0x02	subfunction code	SFUNC_RADIO
6	VAR	message length	LSB
7	VAR	message length	MSB

The data within this packet returns in an ordered fashion, the 12 defined frequencies used for the remote-to-transceiver and transceiver-to-transceiver radio channels. The 2 primary data structures are defined below.

The CHANNEL data structure contains, in Byte 0, the channel number currently in use along with its signal quality index in bytes 1-2. The higher the signal quality value, the quieter the channel.

The SETRADIO data structure contains the 12 defined frequencies in a CHANNEL type array. The first six defined channels are for remote-to-transceiver radio, and the latter six channels are for transceiver-to-transceiver. The user selection frequencies are contained in the following two bytes, transceiver-to-remote and transceiver-to-transceiver respectively.

Table 3-38. Get Radio Frequencies Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet

Select New Remote Frequency

Requests SAB assign the specified radio channel as the "in-use" channel for remote-to-transceiver and transceiver-to-transceiver radio communication.

The specified frequency is one of the currently defined 6 frequencies allowed for use. Use the Get Radio Frequency to obtain the set of 6 available frequencies.

Table 3-39. Select New Remote Frequency Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x01	function code	FUNC_SETINFO
5	0x02	subfunction code	SFUNC_RADIO
6	VAR	message length	LSB
7	VAR	message length	MSB

The data field contains two bytes for the selected frequency. An error is generated if the frequency selected is not one of the currently defined frequency sets. Five minutes before a frequency change, the system notifies all transceivers and listening terminals. If only one frequency field is changed, others can be set to zero. Byte 0 is the transceiver-to-remote channel and byte 1 is the transceiver-to-transceiver channel.

Table 3-40. Select New Remote Frequency Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-41
2	0x00	data length	
3	0x00	data length	
4	0x01	function code	FUNC_SETINFO
5	0x02	subfunction code	SFUNC_RADIO
6	0x00	message length	
7	0x00	message length	

No data field returns with this response.

Table 3-41. Select New Remote Frequency Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x02	BAREER_BADDATA	invalid channel specified

Delete Specific Transceiver Unit

Requests SAB clear the NVM on specified transceiver.

Use this command before removing a transceiver from the Spectrum One network. The transceiver non-volatile memory (NVM) contains frequencies, chipping sequences, and transceiver numbers that are invalid in another network. Transceivers operate with old unit numbers until physically removed from the network or reset.

Table 3-42. Delete Specific Transceiver Unit Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x01	function code	FUNC_SETINFO
5	0x22	subfunction code	SFUNC_DELBASES
6	VAR	message length	LSB
7	VAR	message length	MSB

The data field contains the transceiver unit number selected for removal from the configuration. The data is formatted as an unsigned-integer (2 bytes) from 3 to 63.

Table 3-43. Delete Specific Transceiver Unit Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-44
2	0x00	data length	
3	0x00	data length	
4	0x01	function code	FUNC_SETINFO
5	0x22	subfunction code	SFUNC_DELBASES
6	0x00	message length	
7	0x00	message length	

No data field returns with this response.

Table 3-44. Delete Specific Transceiver Unit Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x02	BAREER_BADDATA	invalid transceiver number

Restart SAB

Requests SAB restart with warm boot. The host is unaffected except losing session connection to SAB. The SAB responds before executing the warm boot.

A message indicating time and SAB version is sent to the host when the SAB is active again.

Table 3-45. Restart SAB Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	0x00	data length	
3	0x00	data length	
4	0x01	function code	FUNC_SETINFO
5	0x23	subfunction code	SFUNC_RESTART
6	0x00	message length	
7	0x00	message length	

No data field is sent with this packet.

Table 3-46. Restart SAB Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-47
2	0x00	data length	
3	0x00	data length	
4	0x01	function code	FUNC_SETINFO
5	0x23	subfunction code	SFUNC_RESTART
6	0x00	message length	
7	0x00	message length	

No data field returns with this response.

Table 3-47. Restart SAB Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet

Get Current SAB Version

Requests SAB return SAB software version as an ASCII NULL-terminated string.

Table 3-48. Get Current SAB Version Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	0x00	data length	
3	0x00	data length	
4	0x00	function code	FUNC_GETINFO
5	0x26	subfunction code	SFUNC_VERSION
6	0x00	message length	
7	0x00	message length	

No data field is sent with this packet.

Table 3-49. Get Current SAB Version Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-50
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x00	function code	FUNC_GETINFO
5	0x26	subfunction code	SFUNC_VERSION
6	VAR	message length	LSB
7	VAR	message length	MSB

Table 3-50. Get Current SAB Version Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet

Get Topology

Requests system topology from SAB. System topology gives information for displaying, in graphical format, all transceivers and remote assignments within the Spectrum One network.

If the data returned is more than one packet, successive packets are sent until all data is sent. The response packet identifies whether response is complete or if more data is being sent.

Table 3-51. Get Topology Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	0x00	data length	
3	0x00	data length	
4	0x00	function code	FUNC_GETINFO
5	0x90	subfunction code	SFUNC_TOPOLOGY
6	0x00	message length	
7	0x00	message length	

No data field is sent with this packet.

Table 3-52. Get Topology Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1 not used bit 2 = 0 EOT (no more packets) = 1 More data to send bit 3-7 not used
1	VAR	return code	Refer to Table 3-53
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x00	function code	FUNC_GETINFO
5	0x90	subfunction code	SFUNC_TOPOLOGY
6	VAR	message length	LSB
7	VAR	message length	MSB

The Get Topology command often returns more than a single packet. Bit 2 in the flags field indicates that more data exists in successive packets. This bit is set for all packets except the last.

Data returned from a Get Topology command contains a linear list of 2 or more topology table structures. It contains one topology table structure for each unit (pseudo transceiver, hardware transceiver, terminal) in the topology. There are always two pseudo transceivers and two or more topology table structures returned. Each 41-byte table represents a node.

The topology table structure is defined as:

```
typedef struct//  Topology table
{
  BYTE     bLevel;       //  Display Level
  UNITTYPE utType;       //  Unit type
  USHORT   usUnit;       //  Unit number
  USHORT   cusChildren;  //  Number of children
  USHORT   ausLevel[16]; //  Unused
  MEDIA    media;        //  Media type

} TOPTABLE;
```

bLevel an integer between 0 and 3 indicating the display level of the network element. Corresponds to screen column that the element description string is typically printed in.

0 pseudo-transceivers
1 primary-transceiver or remotes-in-cradle
2 coax-transceivers or remotes-attached-to-primary-transceiver
3 remotes-attached-to-coax-transceiver

utType unit type defined as:

```
typedef enum//  Unit types
{
   UT_UB,   //  Uninitialized transceiver
   UT_PB,   //  Pseudo-transceiver
   UT_HB,   //  Hardware transceiver
   UT_UR,   //  Uninitialized remote
   UT_RA,   //  Remote on Air
   UT_RC,   //  Remote in Cradle
   UT_NUM   //  Number of enums

} UNITTYPE;
```

usUnit network element unit number.

cusChildren number of downstream units that a particular unit, such as a transceiver, has allocated to it.

ausLevel[] unused empty array.

media indicates the media type associated with the unit defined as:

```
typedef enum   //  Connection media
{
   MED_UNINIT, //  Uninitialized transceiver
   MED_PSEUDO, //  Pseudo-transceiver
   MED_PLUGIN, //  Plug-in transceiver
   MED_SERIAL, //  Serial (RS-232/422)
   MED_COAX,   //  Coaxial cable
   MED_RF,     //  Radio
   MED_NUM     //  Number of MEDIA types

} MEDIA;
```

The topology table structure order within the data returned from the Get Topology command is important. The order is the same as the line-by-line unit listing as on a topology listing. The pseudo-code for generating the topology data might be:

```
Recurse( unit )
{
  output_top_structure( unit );

  for( loop = 1 to unit -> cusChildren )
  {
        Recurse( child[ loop ] );
  }
}

main()
{
  Recurse( pseudo_transceiver_1 );
  Recurse( pseudo_transceiver_2 );
}
```

Table 3-53. Get Topology Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet
0x05	BARERR_OPERATIO N	not enough resources to process command

Get Statistics

Requests SAB return Spectrum One network statistics. System statistics provides the packet count and DCR errors detected within sampling time frame.

Table 3-54. Get Statistics Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x40	command code	BAR_ADMIN
2	0x00	data length	
3	0x00	data length	
4	0x00	function code	FUNC_GETINFO
5	0x0A	subfunction code	SFUNC_STATS
6	0x00	message length	
7	0x00	message length	

No data field is sent with this packet.

Table 3-55. Get Statistics Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-56
2	VAR	data length	LSB
3	VAR	data length	MSB
4	0x00	function code	FUNC_GETINFO
5	0xA0	subfunction code	SFUNC_STATS
6	VAR	message length	LSB
7	VAR	message length	MSB

The data returned is a set of long integers (4 bytes each) detailing the statistics. The following is a mnemonic description of each field.

Bytes 00-03	ErrInternal	Internal errors
Bytes 04-07	CtrlRcvd	Control packets received
Bytes 08-11	CtrlSent	Control packets sent
Bytes 12-15	PaksToNet	Packets sent to network
Bytes 16-19	PaksFromNet	Packets received from network
Bytes 20-23	DcrErrs	DCR errors reported

Table 3-56. Get Statistics Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet

Clear Statistics

Requests SAB clear Spectrum One network statistics and zero all statistics structure fields.

Table 3-57. Clear Statistics Command Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 1 command packet bit 1-7 reserved
1	0x04	command code	BAR_ADMIN
2	0x00	data length	
3	0x00	data length	
4	0x01	function code	FUNC_SETINFO
5	0xA0	subfunction code	SFUNC_STATS
6	0x00	message length	
7	0x00	message length	

No data field is sent with this packet.

Table 3-58. Clear Statistics Response Packet

Byte	Value	Description	Details
0	VAR	flags	bit 0 = 0 response packet bit 1-7 reserved
1	VAR	return code	Refer to Table 3-59
2	0x00	data length	
3	0x00	data length	
4	0x01	function code	FUNC_SETINFO
5	0xA0	subfunction code	SFUNC_STATS
6	0x00	message length	
7	0x00	message length	

No data field returns with this response.

Table 3-59. Clear Statistics Return Codes

Value	Name	Description
0x00	BAR_RTN_OK	good return
0x01	BARERR_BADPAK	invalid packet

Command Code Values

Command, function, sub-function and return code values are as follows:

Table 3-60. Command Codes

Value	Name	Description
0x40	BAR_ADMIN	Administrative Request

Table 3-61. Function Codes

Value	Name	Description
0x00	FUNC_GETINFO	Get Administrative Information
0x01	FUNC_SETINFO	Set Administrative Information

Table 3-62. Sub-Function Codes

Value	Name	Description
0x02	SFUNC_RADIO	Radio Frequencies
0x20	SFUNC_UNITS	Remote and Transceiver Unit Numbers
0x21	SFUNC_DELUNITS	Remove Remote Terminals
0x22	SFUNC_DELBASES	Clear Transceiver NVM Memory
0x23	SFUNC_RESTART	Restart SAB
0x24	SFUNC_UPDBR	SAB Software
0x25	SFUNC_LOGLVL	Logging Level
0x26	SFUNC_VERSIONS	SAB Version
0x90	SFUNC_TOPOLOGY	SAB Topology
0xA0	SFUNC_STATS	SAB Statistics

Table 3-63. Administrative BAR Command Summary

Command	Command Code	Function Code	Sub-Function Code
Get Radio Frequencies	0x40	0x00	0x02
Select New Remote Frequency	0x40	0x01	0x02
Get Units	0x40	0x00	0x20
Delete Specific Remote Units	0x40	0x01	0x21
Delete Specific Transceiver Unit	0x40	0x01	0x22
Restart SAB	0x40	0x01	0x23
Select SAB Software Update	0x40	0x01	0x24
Get SAB Software Update Information	0x40	0x00	0x24
Get Debug Logging Levels	0x40	0x00	0x25
Set Debug Logging Levels	0x40	0x01	0x25
Get Current SAB Version	0x40	0x00	0x26
Get Topology	0x40	0x00	0x90
Get Statistics	0x40	0x00	0xA0
Clear Statistics	0x40	0x01	0xA0

Appendix A

Hex Image Download

The hex image stored in the terminal *non-volatile memory (NVM)* may need replacement if the hex file is corru pt or a custom hex file is developed. Hex image download requires:

- File Access BARs are available on the host and enabled on the SAB
- the loader program is available on terminal
- the hex image is available on host in a directory where the SAB requests files
- the cradle is attached to the SAB.

Note: The enablers offer support for file management support when using the SAB. Refer to the Enabler Programmer's Reference Manual.

To download the hex image from the host to the terminal:

1. From the terminal, run the loader program. At the DOS prompt, enter:

   `loader` *`filename`*

   where *filename* is the name of the hex image file. The program assumes the .HEX extension.
2. At the protocol selection screen, select Spectrum One by pressing UP-ARROW and DOWN-ARROW and press ENTER.
3. Press ENTER from the terminal to erase NVM.
4. Insert terminal into cradle. The terminal NVM is erased.
5. Press ENTER to continue download.
6. After download is complete and successful, press ENTER to warm boot terminal. The terminal should boot with the new hex image.

PAGE INTENTIONALLY BLANK

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

70-13151-01