Version 1.5

Iridium Short Burst Data
Interface Control Document

APPROVED:

_____ _____

Mark Adams

CTO, ISLLC

_____ _____

Revision History

Version	Date	Author	Reason
0.1	10-Jun-2002	Handset Team	First draft
0.2	12-Jun-2002	Handset Team	First Review
0.3	19-Jun-2002	Handset Team	Second Review
0.4	3-Jul-2002	Handset Team	Marketing / Business Development Review
0.5	11-Jul-2002	Handset Team	Final Review
1.0	26-Jul-2002	Handset Team	First customer version
1.1	15-Aug-2002	Handset Team	Integrated back end interface, changed document title
1.2	2-Sep-2002	Handset Team	First review of integrated document
1.3	18-Sep-2002	Handset Team	Second review of integrated document
1.4	20-Sep-2002	Handset Team	Second customer version
1.5	26-Sep-2002	Handset Team	Third customer version

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the protocol for communication between the data terminal equipment (DTE) Field Application (FA), the Short Burst Data (SBD) Subscriber Device (SSD), and the Vendor Application (VA). This document is primarily intended for use by the engineers who are interfacing to the SSD.

1.2 Scope

This document defines all the AT commands/responses used in the protocol and defines how the protocol works through an overview and command descriptions. A basic description of the physical interface requirements are provided at the end of this document. Additional details of the physical interface and complete list of AT commands can be found in [2] and [3] respectively.

1.3 Definitions, Acronyms, and Abbreviations

API	Application Programming Interfaces
ATC	AT Command
DTE	Data Terminal Equipment
ESS	ETC SBD Subsystem
ETC	Earth Terminal Controller
FA	Field Application
Forward	The process of passing a message to the end recipient.
IMEI	International Mobile Equipment Identity
LBT	L-band Transceiver
MIME	Multipurpose Internet Mail Extensions
MO	Mobile Originated
MOMSN	Mobile Originating Message Sequence Number
MSN	Message Sequence Number
MT	Mobile Terminated
MTMSN	Mobile Terminating Message Sequence Number
Receive	The process of an end device taking delivery of a message. Only the SSD and the vendor application can receive a message. The ESS never receives a message it only stores it.

Receive Buffer	This is the buffer in which an SBD message sent from the ESS to the SSD will be stored.
SBD	Short Burst Data
SCADA	Supervisory Control and Data Acquisition
Send	The process of creating a message to be transferred to another device. Only the SSD and the vendor application can send a message. The ESS never sends a message, it only forwards a message.
Send Buffer	This is the buffer in which an SBD message to be sent from the SSD to the ESS will be stored.
SSD	SBD Subscriber Device
Store	The process of holding a message until the message can be forwarded.
TBR	To be Reviewed
UTC	Coordinated Universal Time
VA	Vendor Application

1.4 References

[1] Short Burst Data Service Interface Control Document, version 0.7

[2] Sebring Interface Control Document, version 1.0

[3] ISU AT Command Reference (SSP-ISU-CPSW-USER-0005), version 1.7

2 Overview

Short Burst Data (SBD) is a simple and efficient transport capability used to transfer messages slightly less than 2 kilobytes from field equipment to a centralized information recipient connected to the Internet or other similar type network. This service also supports the transfer of messages from a centralized information source to the field equipment.

The primary segments of the SBD architecture are a DTE FA, the SSD, the Iridium satellite network, the Earth Terminal Controller SBD Subsystem (ESS) located at the Iridium gateway, the Internet, and the vendor application. The relationships between these various segments can be seen in Figure 1. The DTE FA can be any remote monitoring field equipment that needs to send and/or receive messages to a centralized information recipient/source. The SSD is a standard Iridium 9522 L-band transceiver (LBT) with the SBD feature activated. The ESS is responsible for storing and forwarding messages from the SSD to the vendor application and storing messages from the vendor application to forward to the SSD. The SSD communicates with the ESS via the Iridium satellite network.

The interface between the vendor application and the ESS uses standard Internet mail protocols to send and receive messages. Messages are sent to the ESS using a common email address, identifying the SSD using a unique SSD IMEI in the subject line. The message itself is transported as an attachment. Messages sent to the vendor application are delivered to a specific email address which is configured when the SSD is activated.

The interface between the DTE FA and the SSD is an AT command driven interface which is used to load and retrieve messages between the SSD and the DTE device. In the normal flow of events a message is loaded into the SSD (+SBDWB or +SBDWT), a message transfer session between the SSD and the ESS is initiated (+SBDI), and finally if a message was received from the ESS it will be retrieved by the DTE FA (+SBDRB or +SBDRT). Message transfers between the SSD and the ESS use a reliable transport mechanism that ensure the message is delivered error free. If the SSD was not able to send or receive messages, an indication is passed to the DTE FA.

The send and receive buffers in the SSD will maintain messages as long as the SSD is powered on. Once a message is transferred from the DTE FA to the send buffer in the SSD, it will remain there even after it is successfully sent to the ESS. If a message is received at the SSD from the ESS, it will remain in the receive buffer even after the DTE FA reads it. The buffers in the SSD will be cleared only when given an explicit command (+SBDD) or when the SSD is power cycled.

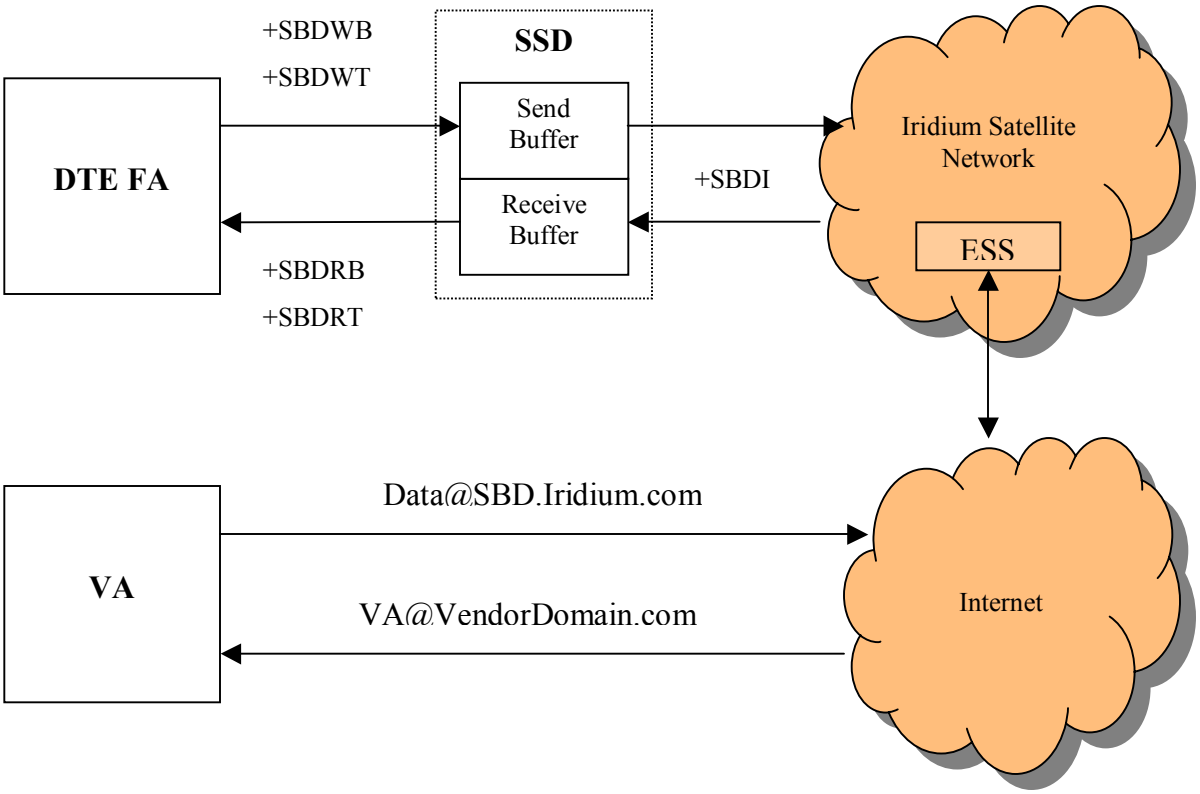

Figure 1 SBD Segments

3 VA to ESS Interface

3.1 Mobile Terminated Messages

In order to send a message from the vendor application to the DTE FA, the vendor application must send the message to the ESS where it will be queued for delivery awaiting contact from the SSD to retrieve it. Figure 2 provides an example MT email message, the formatting rules are outlined below.

- Messages to all SSD units are sent to the email address: Data@SBD.Iridium.com
- Placing the SSD IMEI into the subject line of the email identifies the destination SSD unit.
- If the message does not contain a valid sender (From address), the message will be dropped.
- The message to the SSD will be carried in the attachment to the email.
- The attachment name must have a .sbd extension.
- The attachment must use standard Multipurpose Internet Mail Extensions (MIME) Base64 encoding as defined in RFC 2045.
- The message body plays no role in the message transfer process; any information contained in the body will be discarded.

```
To: Data@SBD.Iridium.com
From: VA@VendorDomain.com
Subject: 304050607080903
<Message Attached>
```

Figure 2 Mobile Terminated Email Message

If the vendor application attempts to queue more than 50 messages for delivery at the SSD, a rejection notice email similar to Figure 3 will be sent to the message originator (From address).

```
To: VA@CustomerSite.com
From: <Iridium SBD Service (Tempe, AZ)>
Subject: Failure to send to unit 304050607080903
Original message is attached
<Message Attached>
```

Figure 3 Mobile Terminated Email Message Rejection Notice

3.2 Mobile Originated Messages

Messages sent from the SSD to the ESS are stored at the ESS then forwarded immediately to the destination address configured when the SSD unit was activated. The message sent to the vendor application from the SSD will be carried in the attachment to the email, which has been encoded using standard MIME Base64 encoding as defined in RFC 2045. Unlike messages sent to the ESS, messages sent to the vendor application carry user data in the message body. The data includes the mobile originated message sequence number (MOMSN), the time of the session, the session status, the message size, and the SSD unit location. The format of the email message is provided in Figure 4, details of the email message fields are provided in Table 1.

```
From: <Iridium SBD Service (Tempe, AZ)>
Sent: Tuesday, August 13, 2002 16:51 PM
Subject: SBD Msg From Unit: 304050607080903

MOMSN: 2
MTMSN: 239
Time of Session (UTC): Tue Aug 13 16:51:04 2002
Session Status: TRANSFER OK
Message Size (bytes): 1230

Unit Location: Lat = 59.372463 Long = 75.309806
CEPradius = 3

Message is Attached.
```

Figure 4 Mobile Originated Email Message

Table 1 VA Received Email Message Fields

Field Name	Description
From	This field typically identifies the sender of the email message. In the case of messages sent to the vendor application, this field is filled with the string "<Iridium SBD Service (Tempe, AZ)>" which is an invalid email address.
Sent	This field provides the time at which the message was emailed from the ESS to the vendor application. The timestamp is in UTC.
Subject	This field provides the SSD IMEI of the unit that sent the message.
MOMSN	This field provides the MSN used by the SSD when the message was

	sent from the SSD to the ESS. This value is incremented each time an SBD session is successfully completed between the SSD to the ESS. This wrap around counter can range from 0 to 65535.
MTMSN	This field provides the MSN used by the ESS when the message was forwarded from the ESS to the SSD. This field will only appear if a message was forwarded to the SSD. This value is incremented each time the ESS forwards a message to the SSD. This wrap around counter can range from 0 to 65535.
Time of Session	This field provides a UTC timestamp of the SSD session between the SSD and the ESS.
Session Status	This field provides an indication of success of the SBD session between the SSD and the ESS. The specific values of the Session Status can be found in Table 2.
Message Size	This field provides an indication of the size of the attached message in decoded form. This not the length of the MIME encoded data.
Unit Location	This field provides the geographic location of the SSD. The latitude and longitude provide a center point and the CEPradius (measured in kilometers) provides the radius of a circle around the center point. The reported position is accurate (within the reported circle) 80 percent of the time.
CEPradius	This field provides an estimate of the accuracy of the unit location. The latitude and longitude provide a center point and the CEPradius (measured in kilometers) provides the radius of a circle around the center point. The reported position is accurate (within the reported circle) 80 percent of the time.

Table 2 Session Status Values

Session Status	Description
TRANSFER OK	The SBD session between the SSD and the ESS completed successfully.
INCOMPLETE CALL	The SBD session between the SSD and the ESS did not complete due to a protocol error. This would imply a problem with the Iridium network.
SBD DENIAL	The SSD is not allowed to access the system
SBD TIMEOUT	The SBD session between the SSD and the ESS did not complete for an unknown reason such as an RF fade.

4 DTE FA to SSD Interface

4.1 AT Commands

This section describes the subset of AT commands that implement the SBD functionality. Refer to [3] for all other AT commands that are supported by the Iridium equipment.

4.1.1 +SBDDSC – Set Delivery Short Code

This command sets the value of the delivery short code (DSC) that will be used for all transactions with the ESS. The DSC will be used to route messages to various locations by assigning numbers to different destinations such as email addresses or co-located servers. This feature is not yet supported at the ESS and should not be changed from the default value of zero.

Set Command: +SBDDSC=[<DSC Value>]<CR>

- This field will default to zero at power-up.
- <DSC Value> has a valid range of 0 to 255.

Set Command Response:

0: Value of the DSC successfully changed.

1: The value of the DSC is out of the allowable range.

Read Command: +SBDDSC?<CR>

Read Command Response:

+SBDDSC: <DSC Value><CR>

4.1.2 +SBDWB – Write binary data to the SSD

This command is used to transfer a binary SBD message from the DTE FA to the single send buffer in the SSD.

Exec Command: +SBDWB=[<SBD message length>]<CR>

- Once the command is entered, the SSD will indicate to the DTE FA that it is prepared to receive the message by sending the ASCII encoded string "READY<CR><LF>" (hex 52 45 41 44 59 0D 0A) to the DTE FA.
- The <SBD message length> parameter represents the length, in bytes, of the SBD message not including the mandatory two-byte checksum.
- The maximum send SBD message length is 1960 bytes. The minimum send SBD message length is 1 byte.
- Once the DTE FA receives the READY indication from the SSD, the SBD message must be sent from the DTE FA formatted as follows:

  {binary SBD message} + {2-byte checksum}

- The checksum is the least significant 2-bytes of the summation of the entire SBD message. The high order byte must be sent first. For example if the DTE FA were to send the word "hello" encoded in ASCII to the SSD the binary stream would be hex 68 65 6c 6c 6f 02 14.
- The send buffer will be empty upon power-up.

- If any data is currently in the send buffer, it will be overwritten.

Command Response:

0: SBD message written to SSD successfully.

1: SBD message write timeout. An insufficient number of bytes were transferred to SSD during the transfer period of 60 seconds.

2: SBD message checksum sent from DTE does not match the checksum calculated at the SSD.

3: SBD message size is not correct. The maximum send SBD message length is 1960 bytes. The minimum send SBD message length is 1 byte.

4.1.3 +SBDRB – Read binary data from SSD

This command is used to transfer a binary SBD message from the single receive buffer in the SSD to the DTE FA.

Exec Command: +SBDRB<CR>

- The SBD message is transferred formatted as follows:

  {2-byte message length} + {binary SBD message} + {2-byte checksum}

  - The {2-byte message length} field represents the length, in bytes, of the SBD message not including the length field or the mandatory two-byte checksum. The high order byte will be sent first.
  - The maximum receive SBD message length is 1890 bytes.
  - The checksum is the least significant 2-bytes of the summation of the entire SBD message. The high order byte will be sent first. For example if the SSD were to send the word "hello" encoded in ASCII to the DTE FA the binary stream would be hex 00 05 68 65 6c 6c 6f 02 14.
  - If there is no received message waiting to be retrieved from the SSD, the message length and checksum fields will be zero.

- The receive message buffer will be empty upon power-up.

Command Response:

There are no response codes generated by the SSD for this command.

4.1.4 +SBDWT – Write a text message to the SSD

This command is used to transfer a text SBD message from the DTE FA to the single send buffer in the SSD.

Exec Command: +SBDWT=[<text message>]<CR>

- The length of <text message> is limited to 120 bytes, this is due to the length limit on the AT command line interface.
- The message is terminated when a carriage return is entered.
- The send buffer will be empty upon power-up.

Command Response:

OK: SBD message stored in send buffer successfully.

ERROR: An error occurred storing SBD message in send buffer

4.1.5 +SBDRT – Read a text message from the SSD

This command is used to transfer a text SBD message from the single receive buffer in the SSD to the DTE FA. This command works just like the +SDDRB but does not provide a length indicator or checksum. The intent of this command is to provide a human friendly interface to SBD for demonstrations and application development.

Exec Command: +SBDRT<CR>

- Once the command is entered, the SBD message in the receive buffer is sent out of the port.
- This command is similar to the +SBDRB except no length or checksum will be provided.

  The maximum receive SBD message length is 1890 bytes.
- The receive message buffer will be empty upon power-up.

Command Response:

+SBDRT: <CR> {receive buffer}

4.1.6 +SBDI – Initiate an SBD session

This command initiates an SBD session between the SSD and the ESS. If there is a message in the send buffer it will be transferred to the ESS. Similarly if there is one or more messages queued at the ESS the oldest will be transferred to the SSD and placed into the receive buffer.

Exec Command: +SBDI<CR>

- The message, if any, in the send buffer will be sent from the SSD to the ESS.

- If there is a message queued at the ESS it will be transferred to the SSD and placed into the receive buffer.

Command Response:

+SBDI: <send status>,<MOMSN>,<receive status>,<MTMSN>,<receive length>,<receive queued>

where:

<send status>:

send session status provides an indication of the disposition of the message send transaction. The field can take on the following values:

0: No SBD message to send from the SSD.

1: SBD message successfully sent from the SSD to the ESS.

2: An error occurred while attempting to send SBD message from SSD to ESS.

<MOMSN>:

The Mobile Originated Message Sequence Number (MOMSN) is a value assigned by the SSD when sending a message to the ESS. This value is incremented each time an SBD session is successfully completed between the SSD to the ESS. This wrap around counter can range from 0 to 65535.

<receive status>:

The receive status provides an indication of the disposition of the message receive transaction. The field can take on the following values:

0: No SBD message to receive from the ESS.

1: SBD message successfully received from the ESS.

2: An error occurred while attempting to perform a mailbox check or receive a message from the ESS.

<MTMSN>:

The Mobile Terminated Message Sequence Number (MTMSN) is an SSD specific value assigned by the ESS when forwarding a message to the SSD. This value is indeterminate if the field <receive status> is zero. This value is incremented each time a message is sent from the SSD to the ESS. This wrap around counter can range from 0 to 65535.

<receive length>:

The receive length is the length in bytes of the SBD message received from the ESS. If no message was received, this field will be zero.

<receive queued>:

Receive queued is a count of messages waiting at the ESS to be transferred to the SSD.

4.1.7 +SBDD – Clear SBD Message Buffer(s)

This command is used to clear the send buffer, receive buffer or both.

Exec Command: +SBDD[<Delete type>]<CR>

- The <Delete type> parameter identifies which buffers are cleared.

  0: Clear the send buffer.

  1: Clear the receive buffer.

  2: Clear both the send and receive buffers.

- Using this command or power cycling the phone are the only means by which both buffers are cleared. The receive buffer will be cleared when an SBD session is initiated. Sending a message from the SSD to the ESS does not clear the send buffer; similarly reading a message from the SSD does not clear the receive buffer.

Command Response:

0: Buffer(s) cleared successfully.

1: An error occurred while clearing the buffer(s).

4.1.8 +SBDC – Clear SBD MOMSN

This command will clear (set to 0) the MOMSN number stored in the SSD.

Exec Command: +SBDC

- The message sequence number is maintained even after power cycle.

Command Response:

0: The receive message sequence number was cleared successfully.

1: An error occurred while clearing the message sequence number.

4.1.9 +SBDS – SBD Status

This command returns current state of the send and receive buffers.

Exec Command: +SBDS

Command Response:

+SBDS: <send flag>, <MOMSN>, <receive flag>, <receive sequence number>

where:

<send flag>:

The send flag indicates the existence of a message in the send buffer. The response from the SSD is one of the following numeric codes:

0: No message in send buffer

1: Message in send buffer

<MOMSN>:

The MOMSN identifies the sequence number that will be used during the next send session.

<receive flag>:

The receive flag indicates the existence of a message in the receive buffer. The response from the SSD is one of the following numeric codes:

0: No message in receive buffer

1: Message in receive buffer

<receive sequence number>:

The receive sequence number identifies the sequence number that was used in the most recent receive session. This value will be –1 if there is nothing in the receive buffer.

4.1.10 +SBDTC – Transfer Send Buffer to Receive Buffer

This command will transfer the contents of the send buffer to the receive buffer. Developers of DTE FA can use this command to test reading and writing to the SSD without actually initiating SBD sessions with the ESS.

Exec Command: +SBDTC

Command Response:

0: Send buffer transferred successfully

1: An error occurred while transferring the send buffer.

5 SBD Scenarios

This section will outline some examples of SBD scenarios which will apply the commands outlined in section 4.

5.1 Mobile Origination (MO) Message

In the case that the DTE FA needs to send data to the vendor application, the SSD will be used in a send only fashion. In this scenario, the DTE FA will load a MO message into the SSD, initiate an SBD session, and finally evaluate and act on the results of the SBD session (Table 3). Finally, the ESS will forward the MO message to the vendor application (Figure 5).

Table 3 DTE FA to SSD Interface, MO Message

To SSD (from DTE)	To DTE (from SSD)	Description
AT+SBDWB=351↵		The DTE FA instructs the SSD that it will write a 351 byte message into the SSD.
	READY↵	The SSD informs the DTE FA that it is ready to receive the message
Binary transfer		The DTE FA sends the 351 byte message followed by the two byte checksum to the SSD. This transfer is not echoed.
	0↵	The SSD will send a zero result code to the DTE FA indicating that the message was loaded without error.
AT+SBDI↵		The DTE FA instructs the SSD to initiate an SBD transfer.
	+SBDI: 1, 23, 0, -1, 0, 0↵	The SSD informs the DTE FA that the message was sent successfully using MOMSN 23. No message was received and no messages are queued.
AT+SBDD0↵		The DTE FA instructs the SSD to clear the message from the send buffer.
	0↵	The SSD informs the DTE FA that the message buffer was cleared successfully.

```
From: <Iridium SBD Service (Tempe, AZ)>
Sent: Tuesday, August 13, 2002 12:49 PM
Subject: SBD Msg From Unit: 304050607080903

MOMSN: 23
Time of Session (UTC): Tue Aug 13 16:51:04 2002
Session Status: TRANSFER OK
Message Size (bytes): 351

Unit Location: Lat = 59.372463 Long = 75.309806
CEPradius = 3

Message is Attached.
```

Figure 5 VA to ESS Interface, MO Message

5.2 Mailbox Check / Mobile Terminated (MT) Message

In the case that the DTE FA needs to receive data from the vendor application the SSD will be used in a receive only fashion. The ESS does not have the ability to notify the SSD that a message is waiting so the DTE FA is required to perform a mailbox check by initiating an SBD session. If at least one message is waiting for the SSD at the ESS, the message is received to the SSD. In this case the scenario becomes a receive only case. There is no difference between the mailbox check and the receive only session other than that in the receive only case a message is forwarded from the ESS to the SSD. In this scenario, a MT message is sent from the vendor application to the ESS (Figure 6), the DTE FA will initiate an SBD session, evaluate the results of the SBD session, and finally read the MT message from the SSD (Table 4). After the SBD session completes, the ESS will forward the empty message to the vendor application indicating the disposition of the SBD session (Figure 7).

```
To: Data@SBD.Iridium.com
From: VA@VendorDomain.com
Subject: 304050607080903
<561 byte message attached>
```

Figure 6 VA to ESS Interface, Mailbox Check / MT Message

Table 4 DTE FA to SSD Interface, Mailbox Check / MT Message

To SSD (from DTE)	To DTE (from SSD)	Description
AT+SBDD0		The DTE FA instructs the SSD to clear the send buffer.
AT+SBDI↵		The DTE FA instructs the SSD to initiate an SBD transfer.
	+SBDI: 0, 498, 1, 237, 561, 2↵	The SSD informs the DTE FA that no message was sent and a 561 byte message was successfully received with MTMSN 237. Two additional messages are queued.
AT+SBDRB↵		The DTE FA instructs the SSD to transfer the received message.
	Binary transfer	The SSD sends a two byte length indicator followed by the 561 byte message followed by the two byte checksum to the DTE FA.

```
From: <Iridium SBD Service (Tempe, AZ)>
Sent: Tuesday, August 13, 2002 12:49 PM
Subject: SBD Msg From Unit: 304050607080903

MOMSN: 489
MTMSN: 237
Time of Session (UTC): Tue Aug 13 16:51:04 2002
Session Status: TRANSFER OK
Message Size (bytes): 0

Unit Location: Lat = 59.372463 Long = 75.309806
CEPradius = 3

Message is Attached
```

Figure 7 VA to ESS Interface, Vendor Application Generated Email

5.3 MO and MT Message

In the case that the DTE FA needs to send and receive data from a central information recipient/source the SSD will be used in a send and receive fashion. In this scenario, the vendor application will send the MT message to the ESS (Figure 8), the DTE FA will load the MO message into the SSD, initiate an SBD session, evaluate the results of the SBD session, and read the received message from the SSD (Table 5), finally the vendor application will receive the MO message (Figure 9).

```
To: Data@SBD.Iridium.com
From: VA@VendorDomain.com
Subject: 304050607080903
<429 byte message attached>
```

Figure 8 Vendor Application to ESS Interface, MT Message

Table 5 DTE FA to SSD Interface, Send and Receive

To SSD (from DTE)	To DTE (from SSD)	Description
AT+SBDWB=351↵		The DTE FA instructs the SSD that it will write a 351 byte message into the SSD.

	READY↵	The SSD informs the DTE FA that it is ready to receive the message
Binary transfer		The DTE FA sends the 351 byte message followed by the two byte checksum to the SSD. This transfer is not echoed.
	0↵	The SSD will send a zero result code to the DTE FA indicating that the message was loaded without error.
AT+SBDI↵		The DTE FA instructs the SSD to initiate an SBD transfer.
	+SBDI: 1, 2173, 1, 87, 429, 0↵	The SSD informs the DTE FA that the message was sent successfully using MOMSN 2173. A 429 byte message was received using MTMSN 87. No additional messages are queued.
AT+SBDD0↵		The DTE FA instructs the SSD to clear the message from the send buffer.
	0↵	The SSD informs the DTE FA that the message buffer was cleared successfully.
AT+SBDRB↵		The DTE FA instructs the SSD to transfer the received message.
	Binary transfer	The SSD sends a two-byte length indicator followed by the 429 byte message followed by the two byte checksum to the DTE FA.

```
From: <Iridium SBD Service (Tempe, AZ)>
Sent: Tuesday, August 13, 2002 12:49 PM
Subject: SBD Msg From Unit: 304050607080903

MOMSN: 2173
MTMSN: 429
Time of Session (UTC): Tue Aug 13 16:51:04 2002
Session Status: TRANSFER OK
Message Size (bytes): 351

Unit Location: Lat = 59.372463 Long = 75.309806
CEPradius = 3

Message is Attached.
```

Figure 9 VA to ESS Interface, MO Message

6 9522 LBT Physical Interface

The 9522 LBT has three physical interface connectors. These interfaces include the Multi-Interface connector, the antenna connector, and the SIM connector. These interfaces are used to provide power, RS232 connectivity, antenna connectivity, and SIM access.

6.1 Multi-Interface Connector

The Multi-Interface connector, pictured in Figure 10, is a standard DB25 that is used to provide power and RS232 connectivity. The pin out of the Multi-Interface connector is provided in Table 6.

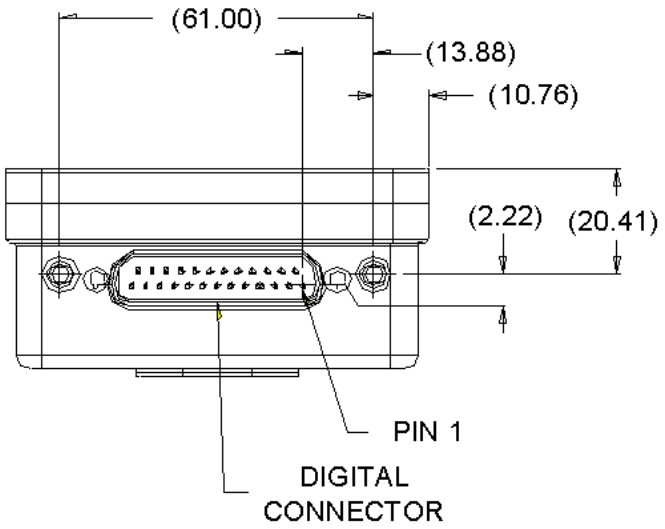

Figure 10 Multi-Interface Connector

Table 6: DC Power, and RS232 Data

Pin	Signal Name	Signal Description	Interface	RS232 DB9 Pin
1	EXT_ON_OFF	Power on/off control input. EXT_ON_OFF control input is pulled to a GND level to toggle the 9522 LBT on and off.	DC Power	
2	Reserved	Reserved	Reserved	
3	GND	External GND input. Same as pin 17, only one needed.	DC Power	
4	EXT_B+	External 4.4 VDC input. Same as pin 16, only one needed.	DC Power	
5	Reserved	Reserved	Reserved	
6	Reserved	Reserved	Reserved	

7	RI	RS232 Ring Indicate.	RS232 Data	9
8	RTS	RS232Request To Send.	RS232 Data	7
9	S_TX	RS232 Transmit Data.	RS232 Data	3
10	DCD	RS232 Data Carrier Detect.	RS232 Data	1
11	Reserved	Reserved	Reserved	
12	Reserved	Reserved	Reserved	
13	S_RX	RS232 Receive Data.	RS232 Data	2
14	GND	Spare GND	Spare GND	
15	Reserved	Reserved	Reserved	
16	EXT_B+	External 4.4 VDC input. Same as pin 4, only one needed.	DC Power	
17	GND	External GND input. Same as pin 3, only one needed.	DC Power	
18	Reserved	Reserved	Reserved	
19	DTR	RS232 Data Terminal Ready.	RS232 Data	4
20	Reserved	Reserved	Reserved	
21	DSR	RS232 Data Set Ready.	RS232 Data	6
22	CTS	RS232 Clear To Send.	RS232 Data	8
23	GND	RS232 ground level signal reference.	RS232 Data	5
24	Reserved	Reserved	Reserved	
25	Reserved	Reserved	Reserved	

6.1.1 DC Power Input Specifications

The DC power requirements for the 9522 LBT are summarized in Table 7 below. Note that these requirements apply to DC power measured at the 9522 LBT multi-interface connector input.

Table 7: DC Power Input Specifications

Parameter	Value
Main Input Voltage - Range	+4.0 VDC to +4.8 VDC
Main Input Voltage - Nominal	4.4 VDC
Main Input Voltage - Ripple	40 mVpp
Peak Input Current (maximum)	3.0 A @ 4.4 VDC
Main Input Active Power (average)	4300 mW
Main Input Standby Power (average)	260 mW

6.2 Antenna Connector

As pictured in Figure 11, the 9522 LBT provides a single, 50 Ohm, TNC type antenna connector for both transmit and receive.

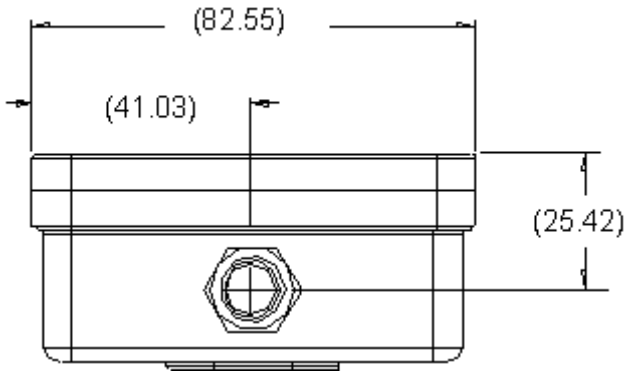

Figure 11 Antenna Connector

6.3 SIM Chip Connector

An integrated SIM chip connector is provided on the 9522 LBT. This connector allows installation of the chip form of the SIM beneath a cover plate on the 9522 LBT housing, as pictured in Figure 12. A valid SIM is required to utilize SBD as well as any other functionality of the Iridium network.

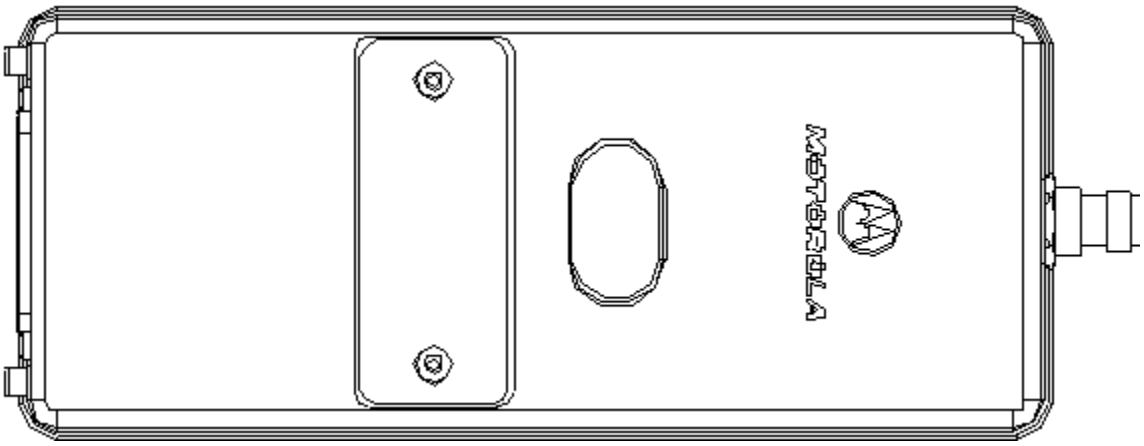

Figure 12 SIM Access Panel

7 Known Issues

This section will identify any problems and issues that have been identified within the system. Issues related to the individual components will be identified in the individual release notes.

- During the beta period any single SSD will be limited to receiving 65535 messages due to a MTMSN rollover issue. If it is necessary to receive more than 65535 messages, Iridium personnel must manually reset the MTMSN.