9210 & XPERT DATALOGGERS

IRIDIUM Manual

Part No. 8800-1193
Version 3.22
February 5, 2018

Table of Contents

Table of Figures

INTRODUCTION

Sutron's Xpert family of DCPs (both the 9210/B and the Xpert/2) have been designed to be easily expandable by adding additional software libraries, called Sutron Link Libraries (SLLs). One such library is Iridium.sll, which adds the ability for the Xpert to communicate over the Iridium network using Short Burst Data (SBD) messages. This document is the user manual for Iridium.sll. The following topics are discussed:

- How to install the library
- What features are supported by the library
- How to configure the Xpert for Iridium communications

INSTALLATION

This section describes installation of the Iridium.sll library.

Installation

To install the Iridium.sll library, copy the file to the "\Flash Disk" subdirectory of your Xpert using Xterm. For more information on performing this file transfer, please refer to chapter 6 of the Xpert or 9210 user manual.

Once the library file has been transferred, reboot the Xpert. The library will load automatically after the Xpert reboots.

To uninstall the library, use Xterm to delete the file from the Flash Disk subdirectory. This can only be done when the Xpert application is not running (select "Exit App" from the Status tab).

In order for the Iridium.sll library to load and operate correctly, the version of the Iridium.sll file must be the same as the version of the application loaded into the Xpert. This is usually not a concern because the same versions of the sll and application are typically packaged together. Should the need arise to verify that the versions are the same, the version of the sll as it resides on the PC can be determined by looking at the file's properties (right-click on the file and select the "Version" tab). The version of the Xpert application is given by the application itself, at the top of the About dialog, which is accessed from the Status tab.

FEATURES

This section discusses what features are supported related to Iridium communications.

- Self-timed (scheduled) Mobile Originated (MO) telemetry data using Short Burst Data (SBD)

- Alarm MO telemetry data using SBD

- Scheduled and manually initiated processing of incoming command messages, i.e., Mobile Terminated (MT) messages

- Set Xpert system time to Iridium system time

- Diagnostic support, including:

    o Scheduling details

    o Next self-timed message content and size

    o Recent incoming messages

    o Signal strength

    o Total message and byte counting

    o Projected data usage

CONFIGURATION

This section discusses how to configure the Xpert to process Iridium messages. There are three different places you'll go to use and configure the Xpert for Iridium communications:

- Iridium Entry - Setup Tab
- ISBD Blocks - Graphical Setup and EzSetup
- Basic Entry – Setup Tab

Iridium Entry - Setup Tab

The Iridium entry on the Setup tab offers modem and scheduling options.
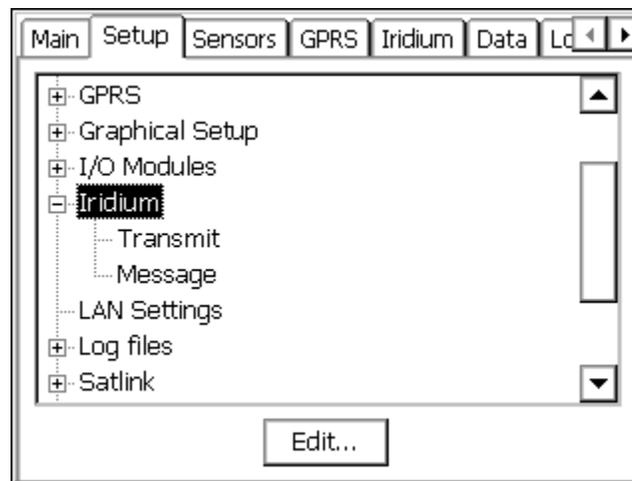
Figure 1: Iridium Entry on Setup Tab

Iridium Properties

To configure the modem and its connection, select the "Iridium" entry and press the "Edit" button. The following dialog will appear.
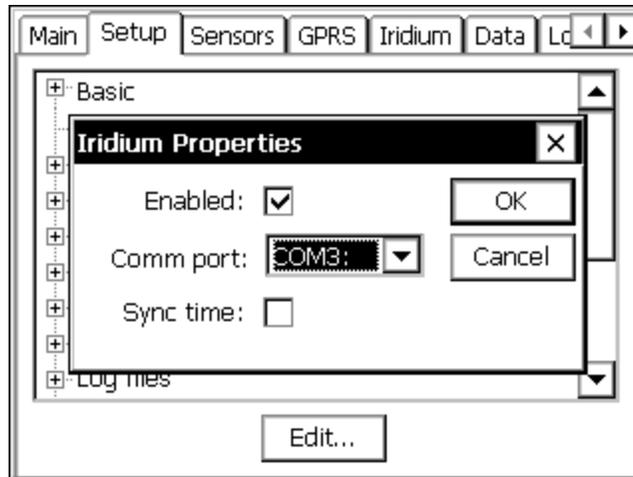
Figure 2: Iridium Properties Dialog

Enabled

The Enabled checkbox determines whether Xpert will do any kind of Iridium processing. The checkbox must be checked for Xpert to initiate self-timed and alarm transmissions, process incoming commands over Iridium, etc. If Enabled is not checked, the Xpert behaves as if the modem is not connected, and will not process any incoming or outgoing transmissions.

Comm Port

Select the Xpert Comm Port to which the Iridium modem is connected.

Sync Time

When the Sync Time checkbox is checked and recording is ON, the Xpert will set its system time to Iridium system time (which follows GMT). This time synchronization will only occur during a transmission or message check, and will only occur up to once daily.

The Iridium system provides GMT time using a counter relative to an "epoch" time defined by the Iridium authority. The epoch time changes roughly every 7 or so years. In order for the Xpert to compute system time properly through a change to the Iridium epoch, the Xpert must store both the current epoch time, and the next expected epoch time. As of version 3.15, the Xpert is programmed to handle an Iridium epoch change in May, 2014.

Iridium will announce new epoch changes in coming years. When a new epoch date is announced, the Xpert must be told to store the new date. To change the value of the next epoch date-time stored in the Xpert, send the following command to the Xpert via Iridium short burst message:

        !irsenxt MM/DD/YYYY HH:MM:SS

For example, to set the next epoch date-time to 5PM June 3, 2020, you would send, "!irsenxt 06/03/2020 17:00:00".

It's also possible to set the *current* Iridium epoch date-time in the Xpert, though this would only be required if the Xpert failed to set the proper epoch date itself during an epoch transition. To set the current epoch date, you would send the command, "!irse MM/DD/YYYY HH:MM:SS".

Again, the irsenxt and irse commands must be received via Iridium short burst data (SBD) message to take effect. These commands won't work if issued over the Xpert command line.

Iridium Transmit Settings

To configure self-timed (scheduled) Mobile Originated (MO) transmissions, select the "Transmit" entry under the Iridium entry on the Setup tab and press the "Edit" button. The following dialog will appear.
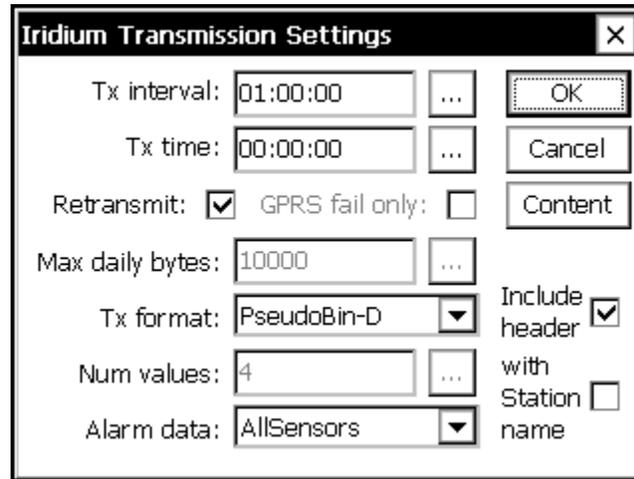
Figure 3: Iridium Transmission Settings

This dialog is used to define the schedule and format of self-timed transmissions, as well as some details about what data to include in alarm transmissions.

Note: use ISBD blocks in the Graphical Setup, and the ISBD option in EzSetup, to define what data to include in transmissions. See ISBD Blocks - Graphical Setup and EzSetup *for more information.*

Tx Interval

Tx Interval defines how often to transmit self-timed messages. The format of the string is, "hh:mm:ss", where "hh" = hours, "mm" = minutes, and "ss" = seconds. For example, "01:00:00" means "every hour", while "00:30:00" means every half hour.

Tx Time

This property defines an offet to Tx Interval. It has the same format as Tx Interval. A Tx Time of "00:15:00" means, "15 minutes after the hour". Some examples of transmission schedules are shown in the table, below:

Tx Interval	Tx Time	Transmission Schedule
01:00:00	00:00:00	12AM, 1AM, 2AM, …, 12PM, 1PM, 2PM, …
02:00:00	00:00:00	12AM, 2AM, 4AM, …, 12PM, 2PM, 4PM, …
01:00:00	00:30:00	12:30AM, 1:30AM, 2:30AM, …, 12:30PM, 1:30PM, …

| 04:00:00 | 00:15:00 | 12:15AM, 4:15AM, 8:15AM, 12:15PM, 4:15PM, … |

Retransmit

When checked, data from failed self-timed transmissions (due to temporary lack of satellite in view, for example), are retransmitted on the next successful self-timed connection. Note that the most recent data is always transmitted first on the first successful reconnect, followed by the oldest to newest data from failed transmissions. Up to 7 days of missed data will be retransmitted.

Note: A system reboot prevents retransmission of data logged by LogField blocks and EzSetup. For complete retransmission support, log data using the standard Log blocks in Graphical Setup.

GPRS Fail Only

When checked, Iridium will only proceed with a scheduled transmission when the on-board GPRS modem fails to indicate it succeeded in its transmission within 3 minutes. This allows you to use Iridium as a back-up to GPRS transmissions, transmitting only when GPRS has failed to transmit its data to the Primary or Secondary servers.

To use Iridium as a back-up to GPRS, check the "GPRS Fail Only" setting, and configure Iridium's Tx Interval and Tx Time to be the same as GPRS's Tx Interval and Tx Time. Also, make sure to include both GPRS and ISBD telemetry tag blocks in the graphical setup to define the content of the transmissions. Since both GPRS and ISBD blocks are required, you have the option to transmit less data over Iridium if desired to reduce data cost.

Max Daily Bytes

When "GPRS Fail Only" is checked, the system will stop making scheduled Iridium transmissions for the day once "Max Daily Bytes" has been met. This limit only applies when "GPRS Fail Only" is checked, and never applies to alarm transmissions, nor when transmitting responses to commands received via SBD message over Iridium.

The value of Max Daily Bytes can be changed by sending a command to the Xpert via Iridium short burst message:

```
!irsmdb nnn
```

For example, to set Max Daily Bytes to 20000, send "!irsmdb 20000".

Include Header

When checked, a header is prepended to the data that describes the type of transmission, as well as details that describe how the message may be broken up over multiple packets. Be sure this option is checked when transmitting to Hydromet Cloud (formerly, "SutronWin").

Include Station Name

When checked, the station name is included in the transmission header. This is typically left unchecked for operation with Hydromet Cloud (formerly, "SutronWin").

Tx Format

This property defines the format of the data sent in self-timed transmissions (as well as alarm transmissions when Alarm Data = "AllSensors"). The following formats are supported:

ASCII Column	ASCII human readable format with sensor values in columns. Each row represents data from a previous measurement interval
BUFR	Binary format per WMO BUFR specification for transmission and storage
PseudoBin-B	6-bit "binary" format, interleaved
PseudoBin-B Non-Int	6-bit "binary" format, non-interleaved
PseudoBin-C	6-bit "binary" format, non-interleaved, with additional meta data
SHEF	ASCII human readable format using SHEF codes
SHEFFIX	ASCII human readable format using SHEF codes, where data appears in fixed width format of 7 chars
NIFC	ASCII human readable format based on NIFC requirements
NFRDS	ASCII human readable format based on NFRDS requirements
Handar	ASCII human readable format
SSP	Sutron Standard Protocol, binary format
Text	ASCII human readable text format, with alarm flags

See Appendix A – Telemetry Formats, for detailed definitions of each of the formats.

Num Values

When the selected format is interleaved (e.g., ASCII Column or PseudoBin-B), the number of data values to include for each selected data item must be the same for all selected items, and so is entered in this dialog (rather than where it is normally entered, in the ISBD telemetry block properties).

For example: if Num Values is 4, and ISBD blocks tag the two data points, "HG" and "HF", then the last 4 values of HG and HF appear in the transmit message.

Alarm Data

Alarm Data can be either "SensorsInAlarm" or "AllSensors". When set to the latter, alarm transmissions contain the same data as is found in self-timed transmissions, i.e., all data tagged with an ISBD block in the graphical setup (as well any data points tagged in EzSetup). However, when Alarm Data is "SensorsInAlarm", only the data *triggering the alarm* is included in the transmission.

Transmit "Content" Dialog

When you press the Content button on the Iridium Transmission Settings dialog, the "Set Tx Content" dialog opens showing a list of all ISBD blocks in the graphical setup (as well as any data points tagged in EzSetup).
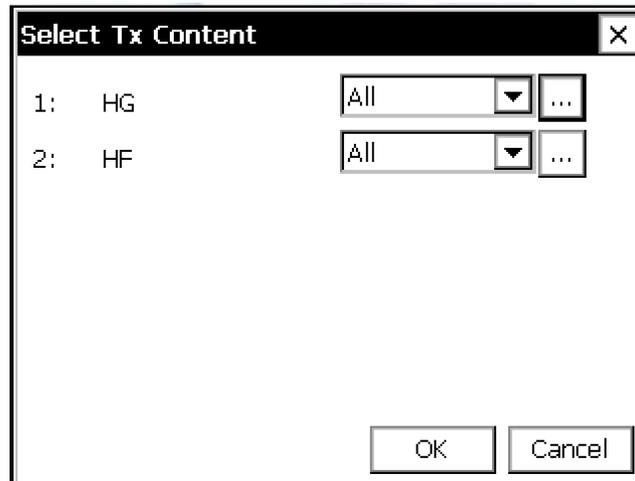
Figure 4: Select Tx Content

When more entries exist than can be shown in the dialog, scroll bars appear, allowing you to scroll to other entries.

From this dialog, you can change the value of the "TxContent" property for each data item in the list. You can also press the "…" button to access and change other properties in the block's properties dialog. See ISBD Block Properties for more information on how to configure these properties.

Mobile Terminated (MT) Messages

When you send a message to the station via Iridium SBD (i.e., a "Mobile Terminated" message), the Xpert will attempt to process the message text as if it's a command meant for the Xpert command-line. All command-line commands are supported. The Xpert sends the entire command line response, in multiple SBD messages, if required (responses are NOT truncated to only a single SBD message).

NOTE: command responses can be LARGE, and can therefore be expensive.

Commands should be preceded by an exclamation point to disable screen pagination, etc. If a data or setup authorization password has been defined, this too must precede the command.

MT Command Examples

!show	Return a listing of sensor and tag values. No authorization required.
!show batt	Show current value of sensor "batt"
!get /today	Show contents of ssp log for today

!info	Show the current system status
!irsenxt MM/DD/YYYY HH:MM:SS	Set next Iridium epoch date
!irse MM/DD/YYYY HH:MM:SS	Set the current Iridium epoch date
!irsmdb nnn	Set Max Data Bytes to nnn (replace nnn with number)
pword !show	Return sensor and tag listing, "pword" required for authorization
pword !reboot	Reboot the station, "pword" required for authorization

See the Xpert or 9210B user manuals for a complete listing of commands.

Configuring Mobile Terminated (Incoming) Message Checks

To configure scheduling of incoming message handling, select the "Message" entry under the Iridium entry on the Setup tab and press the "Edit" button. The following dialog will appear.
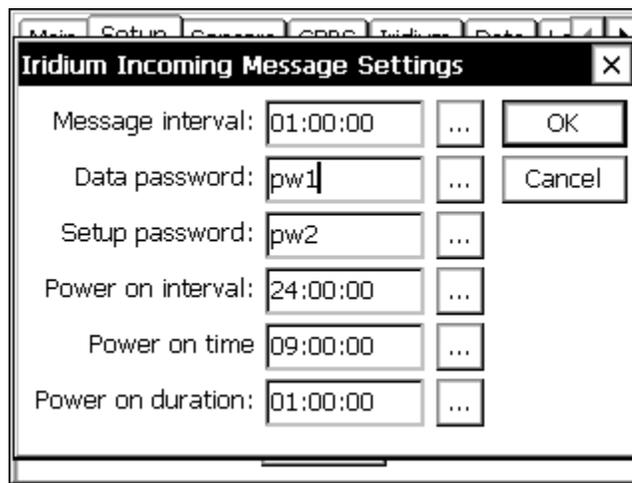
Figure 5: Iridium Message Settings

There are three ways the system can be told to look for and handle incoming messages:

- Scheduled Message Checks
- Powered-on "Listen" Periods
- Manual message check

Scheduled Message Checks

The Message Interval property defines how often to look to see if an incoming message is waiting to be processed. For example, if Message Interval is "01:00:00", then Xpert will look every hour to see if a message is waiting. If a message is waiting, it is retrieved and processed.

Note: scheduled message checks cost money (when not performed as part of a transmission)! Even if a message is not waiting to be processed, you are charged for the message check.

Powered-on "Listen" Periods

During these periods, the modem is powered and will process an incoming message if and when it is received. You are charged for the cost to retrieve the message. The advantage of this method over Scheduled Message Checks is that you won't be charged for message checks when no message is waiting. However, since the modem is powered, more power is being consumed.

The Power on Interval property defines how often to power the modem, while Power on Time defines the offset to this interval, and Power on Duration defines how long to stay on. For example, settings shown in the dialog above will result in a daily period from 9AM to 10AM where the modem is powered and waiting to receive and process messages. Note: any transmissions scheduled during this period still occur.

Manual Message Checks

You can tell the system to check for messages manually from the Iridium diagnostics tab. See the "Check Messages Now" button in the following dialog: Last Messages Received.

Message Check Charges

Note: The system checks for incoming messages on every scheduled and alarm transmission. This message check does NOT cost extra, even if no message is waiting. Hence, the cheapest way to manage incoming messages is to have the system look only when transmissions are made. This eliminates the additional cost of scheduled message checks, as well as the additional power needed for listen periods.

Data Password and Setup Password

These two properties specify authorization codes used to validate incoming command messages. Setup Password specifies the authorization code for "setup-level" access, while Data Password specifies the authorization code for "data-level" access.

Incoming command messages have setup-level access by default. When the Setup Password property contains an authorization string, any and all incoming messages that normally require setup-level access to be processed must prefix the command with the authorization code. For example, to command the station to reboot with an authorization code of "MyAuthCode", the incoming message content would be "MyAuthCode !reboot".

ISBD Blocks - Graphical Setup and EzSetup

Data is selected for inclusion in Iridium transmissions using ISBD blocks in the Graphical Setup, and by checking the "ISBD" option in EzSetup. The ISBD properties are the same, whether you are working in the Graphical Setup or EzSetup.

An example setup is shown below:
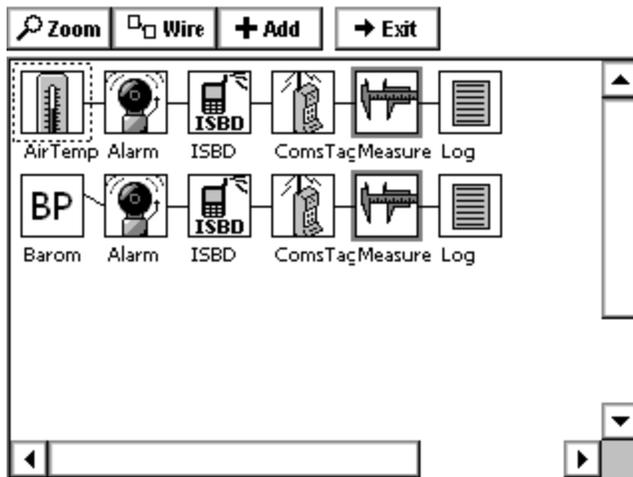
Figure 6: Example ISBD block setup

In the setup depicted above, ISBD blocks "tag" two measured data points, that of, Air Temp and Barom. Here are some important notes about the setup:

- The ISBD block appears after Alarm block. This ensures the ISBD blocks "sees" the result of the alarm evaluation, triggering alarm transmissions, if needed. Of course, the Alarm block is only required if your system will evaluate data for Alarm and/or alarm transmissions.

- The ISBD block appears before the ComsTag block. This ensures the ComsTag "sees" the alarm transmission status generated by the Alarm and ISBD blocks.

ISBD Block Properties

To view and edit the properties of the ISBD block, tap the block and select "Edit Properties" in the pop-up menu that results (in EzSetup, simply press the ISBD button). The following dialog is shown:

Figure 7: ISBD Edit Tag Dialog

The ISBD Edit Tag dialog is used to edit the properties of the ISBD block that "tags" the data point for transmission.

Note, if the selected format is "BUFR", the settings differ slightly than what is shown and described here. Please see BUFR, later in this document, for details.

Label

Label defines the "name" of the data. Some formats (e.g., SHEF) include this Label in the transmission.

Tx Content, Data Interval, Data Time, Num Values

These properties define what data to include in the transmission. See the following table for details:

Tx Content		Description of data to transmit
All		All the data measured since the last transmission (or up to Num Values for interleaved formats), will be transmitted
Individual		Individual values are selected for transmission based on timestamp
	Data Interval	The time interval between readings. For example, "01:00:00" means "hourly data"
	Data Time	The time, or offset into the interval, of the data to include. For example, "00:15:00" means "include data measured on a 15 minute offset"
	Num Values	Number of values for each data item to include in the transmission. Enabled only for non-interleaved transmission formats
Last		Only the last measured value will be transmitted (or the last "Num Values", if enabled in global transmission settings)
Exclude		Exclude this item from transmission

Sequence

This property determines the order of the data item within the transmission.

Right Digits

This property determines the number of right digits to include in the transmission.

Basic Entry – Setup Tab

A Basic program can format custom Iridium messages. Any of the Basic string functions can be used to format the buffer as required, but the Bin and Bin6 functions exist specifically to help forming 8-bit and 6-bit binary encoded data.

To create a self-timed formatting routine, create a function with a "SELFTIMED_" prefix. The return value of the function becomes the buffer. For example:

```
Public Function SELFTIMED_STFormatter
   Selftimed_STFormatter = "Test Selftimed Message"
End Function
```

To create an alarm formatting routine, create a function with an "ALARM_" prefix and a single parameter in which the group number will be passed. The return value of the function becomes the buffer. For example:

```
Public Function ALARM_Formatter(Group)
   ALARM_Formatter = "Test Alarm Reporting Message: " + Str(Group)
End Function
```

It's possible to append to the buffer, as opposed to simply overwriting it, by taking advantage of string concatenation. The following example appends its message to the current buffer:

```
Public Function SELFTIMED_STFormatter
   SELFTIMED_STFormatter = SELFTIMED_STFormatter + "Test Selftimed
Message"
End Function
```

More than one formatting function can exist in your program, but only one self-timed and one alarm function may be active at one time. The active routine is selected in the Iridium Custom Formatting entry under Basic, on the Setup tab:
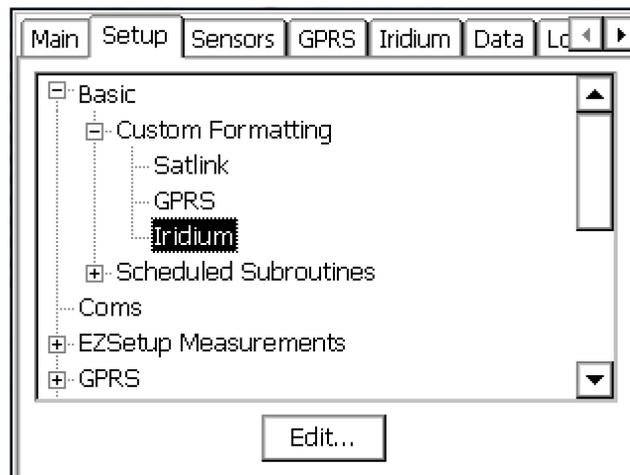
Figure 8: Iridium Custom Formatting Entry

TESTING IRIDIUM OPERATIONS

This section discusses testing your Iridium setup to verify communications and other functions are correct.

Iridium Diagnostics Tab

The Iridium tab contains diagnostic information and tools you can use to verify everything works as desired.
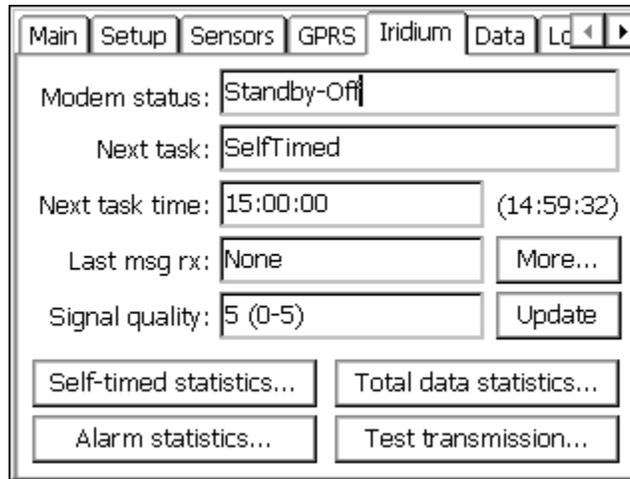
Figure 9: Iridium Diagnostics Tab

Modem Status

Modem status describes the current state of Iridium processing, and may be any one of the values in the following table:

Status	Description
Standby-On	Modem is powered, and is ready for next task
Standby-Off	Modem is not powered, and is ready for next task
SelfTimed Tx	Self-timed transmission in progress
Alarm Tx	Alarm transmission in progress
Message Check	System is checking for presence of incoming message
Ring Alert	Modem has received notice of message waiting from satellite
Power On	Modem is powering on
Power Off	Modem is powering off
Daily Maint	System is storing and resetting daily counts
Syncing Time	System is setting Xpert system time to Iridium system time

Comm Error!	An error has occurred talking to modem (check connection to modem)
Starting	System is starting up
Stopping	System is stopping
Stopped	Iridium processing has stopped (no transmissions or message checks will occur)

Next Task Status

Next task describes the task next scheduled for processing. **Next task time** describes when this task will occur. The time in parenthesis is the current system time.

Last Messages Received

Last msg rx displays the text of the last command received, if any. Press "More…" to see the following dialog, showing the last several messages received, as well as last and next times for power-on/off schedule, if any.
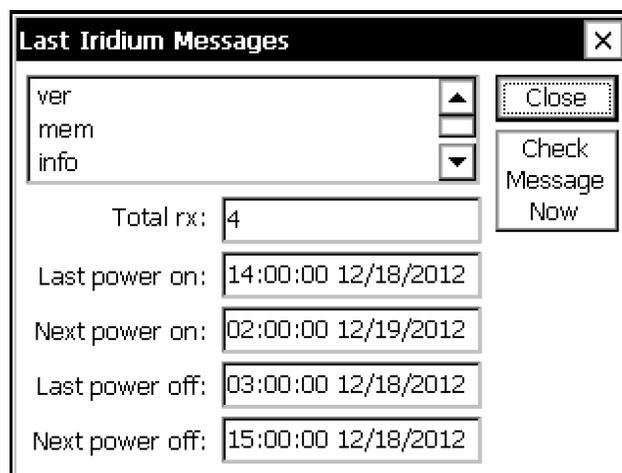
Figure 10: Last Iridium Messages Dialog

Press the "Check Message Now" button to check for pending messages. Note: you will be charged for the message check, in addition to charges for the data of any message received. See Message Check Charges for more information of managing message check costs.

Self-Timed Statistics

Press the "Self-timed statistics" button on the Iridium tab to bring up the following dialog showing detailed information about the current day's self-timed transmissions.
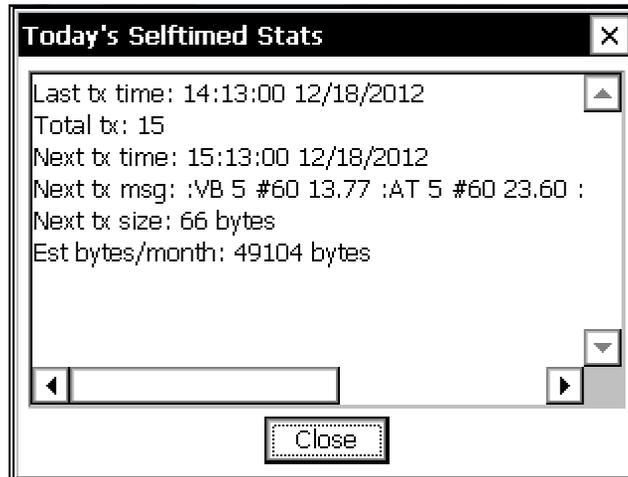
Figure 11: Today's Self-timed Statistics Dialog

Alarm Statistics

Press the "Alarm statistics" button on the Iridium tab to bring up the following dialog showing detailed information about the current day's alarm transmissions.
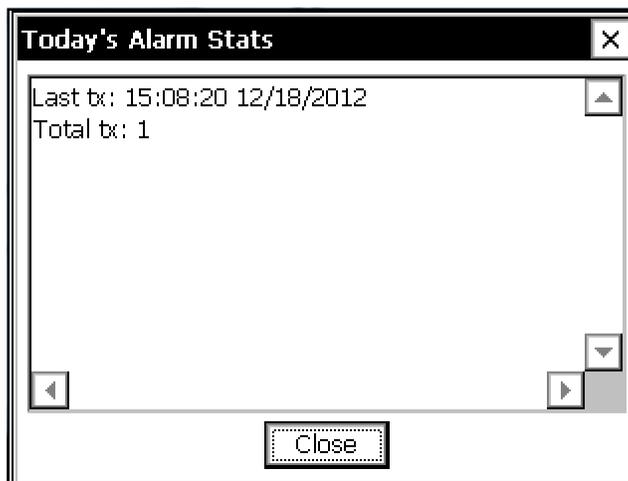
Figure 12: Today's Alarm Statistics Dialog

Total Data Statistics

Press the "Total data statistics" button on the Iridium tab to bring up the following dialog showing the total data sent and received for the current day, so far.
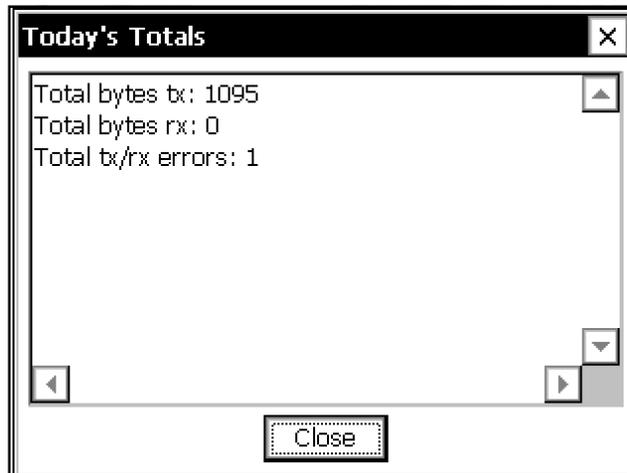
Figure 13: Today's Totals Dialog

Total Tx/Rx Errors

Total tx/rx errors increments any time the system has a problem transmitting or receiving due to a weak signal. Hence, you can use this figure to evaluate how well your antenna is placed. Note that a tx/rx error does NOT mean a transmission totally failed (was never transmitted), but only that an error occurred while trying to transmit. The system continues to retry transmissions experiencing errors, until the transmission finally makes it through.

Totals Over Time

The totals shown on the Iridium diagnostics tab are all daily totals, meaning they apply only to the current day, so far. To track totals over several days, use the system log. At the end of every day, the day's totals are logged to the system log. This allows you to retrieve daily totals, and add them up to see totals over a longer term.

Here's an example log entry of daily totals:

```
12/18/2012,00:00:00,(Stat) - Iridium COM3: Totals: ST=24, AL=0, MT=0,
RA=0, MC=0, Tx=1753, Rx=0, Err=3,,,
```

The following table explains what each entry represents:

Entry	Description
ST	Number of self-timed transmissions for the day
AL	Number of alarm transmissions for the day
MT	Number of received messages for the day
RA	Number of ring alerts (message waiting indications) for the day
MC	Number of message checks for the day
Tx	Total bytes transmitted for the day
Rx	Total bytes received for the day

Err	Total errors occurring during either a transmit or message check attempt

Test Transmission

Press the "Test transmission" button on the Iridium tab to bring up a dialog prompting you for text to transmit. Pressing "Ok" in the dialog transmits the text. Pressing "X" in the dialog cancels the test transmission.
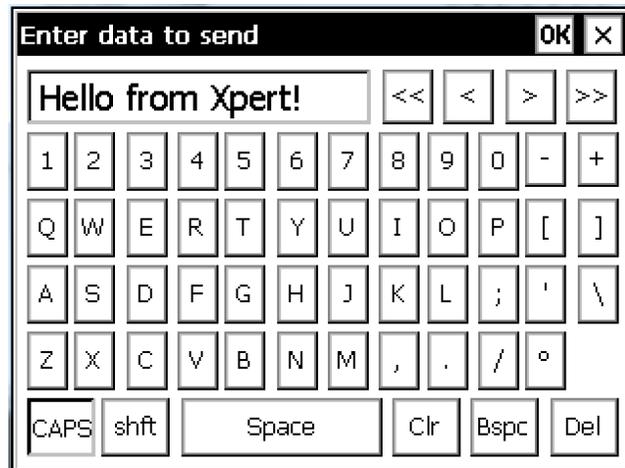
Figure 14: Test Transmission Dialog

Troubleshooting

1. Verify nothing else in the system is using the com port (Coms entry on Setup tab, basic program, modbus, etc.)? Issuing "status" at the command prompt will list all ports that Remote has been configured to support (and are therefore unavailable for Iridium use).

2. Observe the Iridium diagnostics tab to verify that transmissions and message processing are being scheduled as desired, and watch for "Comm Err!", indicating a problem communicating with the modem.

3. Issuing "report high" or "report debug" at the command prompt will provide real time diagnostic information about Iridium performance.

4. The system status page will show information about the most recent diagnostic messages and the current state of the Iridium modem. This can also be displayed on demand at the command prompt with the "info" command.

5. Examine the system.log to look for status messages, connection errors, and transmission errors.

6. Use the Iridium diagnostic tab to send test messages.

7. Contact Sutron Customer Service at 703-406-2800.

APPENDIX A – TELEMETRY FORMATS

This appendix contains descriptions for each of the telemetry formats supported by Xpert Iridium.

ASCII Column Format

The ASCII Column format is an ASCII, human readable format placing sensor values in columns, with each row containing data from earlier measurement intervals (newest data appears in top row).

The following example contains the last 3 measured values for 4 different sensors. Note how "//" is used to indicate "missing" data.

```
-4.422 35.00 136.16 37.00
-4.397 34.00 135.23 36.00
-4.324 33.00 134.88 //
```

The "right digits" property, defined by the ISBD data tag block, determines the number of right digits that appear in the transmission.

The number of rows in the transmission is determined by the "Num Values" property in the "Iridium Transmission Settings" dialog (select "Transmit" entry under the Iridium node on the Setup tab).

Handar Format

The Handar format is an ASCII, human readable format where sensors are separated by <CRLF>, sensor data is separated with a space, and missing data is reported as "M".

For example:

```
010034380517419:21:30G42+0NN155E9200070"
1.3 1.3 1.3 1.3
2.4 2.4 2.4 2.4
M 12.2 12.2 12.2 12.2 12.3 12.3 12.3
```

NFDRS and NIFC Formats

The NFDRS (National Fire Danger Rating System) and NIFC (National Interagency Fire Center) formats produce reports of human readable ASCII values using SHEF codes as labels for data.

The labels applied to measurement outputs are used to identify the data to include in the report. The following table shows what labels are recognized be the NIFC and NFDRS formatter. An "x" means the sensor label must appear in the setup for the formatter to succeed. The items with no "x" are recognized and will be formatted, but are not required.

Sensor	Label	NIFC	NFDRS
Rainfall	PCH	x	x

10-Min Avg Wind Spd	USH	x	x
10-Min Avg Wind Dir	UDH	x	x
Air Temperature	TAH	x	x
Fuel Temperature	MTH	x	
10-Min Avg Rel Hum.	XRH	x	x
Battery Voltage	VBH	x	x
Barometric Pressure	PAH		
Peak Wind Direction	UGX		
Peak Wind Speed	UPH		
Fuel Moisture	MMH		
Solar Radiation	RDH		x

Here's an example NIFC telemetry report:

00.00

000

270

328

110

100

14.0

000

000

020

0486

Here's an example NFDRS telemetry report:

00.00

000

270

328

110

100

14.0

000

000

020

0486

Note how there are no labels in the report. This is because the position of the data within the report determines the source of the data.

Pseudobinary Formats

Pseudobinary formats produce ASCII reports of 6-bit pseudo-binary formatted data values. The formats are "pseudo"-binary, because each sensor value is expressed in the range of ASCII characters, but not in such a way that is readily human-readable.

Pseudobinary B (Interleaved and Non-Interleaved)

The Pseudobinary-B Interleaved format is identical to the 8210 binary transmission format. "Interleaved" means the most recent values of all sensors come first, followed by the next most recent, and so on. "Non-interleaved" means all the data for sensor 1 is followed by all the data for sensor 2, and so on, i.e., the data is not "interleaved" according to time.

This pseudobinary format cannot be easily read by a person. Here's is an example message:

```
B 1 @ @ G t @ S x @ @ i @ G s @ S r @ @ i I
                              └─Battery Voltage
                            └─Temp #2
                        └─Precip #2
                    └─Stage #2
                └─Temp #1
            └─Precip #1
        └─Stage #1
      └─Delta Time
    └─Group ID
  └─Block ID
```


Pseudobinary-B Format

Name	Bytes	Description
Block ID	1	BLOCK-IDENTIFIER is always sent as "B" to indicate that this is the pseudobinary B format.
Group ID	1	GROUP-ID can be "1" to indicate a scheduled transmission, "2" meaning an alarm transmission,

		and "3" indicating a forced transmission.
Delta Time	1	Age in minutes of the most recent data
Data	3x num sensors	Data in either interleaved, or non-interleaved format. The example above shows the data interleaved (most recent of all sensors followed by next oldest, and so on). See the section below, "Six-Bit Binary Encoded Format" for details on how these values are encoded.
Battery Voltage	1	Logger battery voltage measured just prior to transmission

Pseudobinary-C Format

The Pseudobinary-C format also produces an ASCII report of 6-bit pseudo-binary formatted values, but with additional fields to describe label and time information. Just as the "B" version of the format, this format is "pseudo"-binary, because each sensor value and descriptor is expressed in the range of ASCII characters, but not in such a way that is readily human-readable.

Pseudobinary-C Format

Name	Bytes	Description
Block ID	1	BLOCK-IDENTIFIER is always sent as "C" to indicate that this is the pseudobinary C format.
Group ID	1	GROUP-ID can be "1" to indicate a scheduled transmission, "2" meaning an alarm transmission, and "3" indicating a forced transmission.
Measurement Delimiter	1	This byte is always a "+" and it is used to denote the start of measurement data.
Measurement Index (Sequence)	1	The Sequence number assigned by user in the output telemetry settings. This value is encoded 6 bit binary. Typically, the sequence number assigned is unique, allowing you to uniquely identify the data point.
Day	2	This 2 byte encoded 6 bit binary encoded number represents the Julian day of the year. The day tells when the most recent (first) sensor reading of this measurement was made.
Time	2	This 2 byte encoded 6 bit binary encoded number is a number of minutes into the day. It tells when the most recent (first) sensor reading of this measurement was made.
Interval	2	This 2 byte encoded 6 bit binary encoded number

		tells the measurement interval in minutes, or the amount of time between readings of this measurement.
Measurement Data	3 for each sensor reading	Sensor data encoded 6 bit binary.
Additional Measurements	Variable	If more than one measurement was setup for transmission, more data will follow. Each measurement setup will have data starting with the Measurement Delimiter and ending with Measurement Data.
Final Delimiter	1	This byte is always "." and it is used to denote the end of all measurement data.
Battery voltage	1	This is the battery voltage measured prior to making the transmission. The range of the number will be -32 to +31 and can be converted to volts by multiplying by 0.234 and adding 10.6 allowing a range of 3.1 to 18.1 volts.

Example Pseudobinary-C message:

```
C3+ADGTU?///+BDGTU?@UI+CDGTU??~v///.L
Group          3 format C
Sequence       A    1
Julian Day     DG   263
Time           TU   21:41
Interval       ?    00:-1
Data:          ///  missing
Sequence       B    2
Julian Day     DG   263
Time           TU   21:41
Interval       ?    00:-1
Data:          @UI  1353
Sequence       C    3
Julian Day     DG   263
Time           TU   21:41
Interval       ?    00:-1
Data:          ?~v  -74
               ///  missing
Batt V         L    13.4
```

Pseudobinary-D Format

This is another compact data format. It differs from Pseudobinary B in that it has a timestamp at the start of the message. The timestamp indicates when the transmission should have taken place and helps decode when the data was collected. Pseudobinary D is 4 bytes larger than format B.

The timestamp is similar to the one in Pseudobinary C. Pseudobinary D is smaller than Pseudobinary C and it lacks detailed timestamps that would allow one to completely reconstruct the time the data was collected from the message itself. To correctly use Pseudobinary D, the decoder needs to know the measurement setup used.

The benefit of using Pseudobinary D is being able to correctly decode data regardless of when it was sent or received. This allows stations to re-transmit old data and have it correctly interpreted by the decoder while keeping the message size at a minimum.

Name	Bytes	Description
Block ID	1	BLOCK-IDENTIFIER is always sent as "D" to indicate that this is the pseudobinary D format.
Group ID	1	GROUP-ID can be "1" to indicate a scheduled transmission, "2" meaning an alarm transmission, and "3" indicating a forced transmission, or "4" indicating a retransmission
Day	2	This 2 byte encoded 6 bit binary encoded number represents the Julian day of the year. The day tells when the transmission was originally scheduled to take place
Time	2	This 2 byte encoded 6 bit binary encoded number is a number of minutes into the day. It tells when the transmission was originally scheduled to take place
Measurement Data	3 for each sensor reading	Sensor data encoded 6 bit binary. The value transmitted will be value * 10^RightDigits. The string /// will be sent if the data was never measured or had bad quality
Battery voltage	1	This is the battery voltage measured prior to making the transmission. The range of the number will be -32 to +31 and can be converted to volts by multiplying by 0.234 and adding 10.6 allowing a range of 3.1 to 18.1 volts.

Six-Bit Binary Encoded Format

The six bit binary format is used to encode numbers into displayable ASCII characters. Notice that fractional numbers cannot be represented, so for instance a battery voltage of 13.04 volts set up with 2 right digits will be sent as 1304.

- A 1 byte encoded number can range from -32 to +31.

- A 2 byte encoded number can range from -2048 to +2047

- A 3 byte encoded number can range from -131072 to +131071

Binary encoded numbers are always sent most significant bytes first. The number itself is broken down into 6-bit digits, and each digit is placed in one byte of data. The number 64 (ASCII "@") is added to each digit to make it fall within the range of displayable ASCII characters. The only exception is that 127 (ASCII <DEL>) is sent as 63 (ASCII "?")

Example 1. Encoding the number 10 in 1 byte:

Since 10 will fit in 6-bits we only have to add 64 which would yield 74. So the number 10 would appear as ASCII 74 or the letter "J".

Example 2. Encoding the number 12345 in 3 bytes:

First we have to convert 12345 into binary in 6-bit pieces:

      12345 (base 10) = 11 000000 111001 (base 2)

Now we can convert each piece back to base 10:

      11 000000 111001 (base 2) = 3, 0, 57

Finally, we add 64 to each piece and convert to ASCII:

      67, 64, 121 = ASCII "C@y"

Example 3. Encoding the number -12345 in 3 bytes:

First we have to convert -12345 into two's complement 18-bit binary: -12345 (base 10) = 111100 111111 000111 (base 2)

Now we can convert each piece back to base 10: 111100 111111 000111 (base 2) = 60, 63, 7

Finally, we add 64 to each piece and convert to ASCII (since the second piece is 63 we leave it alone):

      124, 63, 71 = ASCII "|?G"

Example 4. Decoding the 3 byte string "@SW":

This is just like encoding except we follow the steps backward.

First we convert all the characters to ASCII decimal codes:

      ASCII "@SW" = 64, 83, 87

Now we subtract 64 from each piece and convert to 6-bit binary:

      0, 19, 23 = 000000 010011 010111

Finally, we combine all the bits to form one 18-bit two's complement number and convert to base 10:

      000000010011010111 = 1239

SHEF and SHEFFIX Formats

"SHEF" format conforms to NESDIS Standard Decimal format specifications. The format is ASCII and readable by persons without the aid of a computer.

The standard decimal format is human readable while binary formats require some processing before the values can be read. The standard decimal format is generally twice as long as the binary transmission. This means that less data can be sent in the same amount of time using the standard decimal format.

An example of a standard decimal format message is shown below. This message comes from a station with three sensors in the self-timed group. The sensors have been named HG, PC and TA. The battery voltage is the voltage of the battery just prior to transmission.

```
:HG 0 #15 10.20 10.15 :PC 0 #15 50 49 :TA 0 #15 -22.
```

```
                                  Precip            Temperature           Battery

                 Older gauge reading (by data interval time)

          Newest gauge reading

     Data Interval

   Offset Time

Gauge Height
```

The names HG, PC, TA and VB used in the transmission are called SHEF Physical Element codes (PE codes) and are the names given to the outputs in the 8310. Be sure to change the output names if you want specific codes sent in the transmission. NESDIS has a recommended list of SHEF PE codes on its website at http://noaasis.noaa.gov/DCS/htmfiles/schefcodes.html

"SHEF Fixed" is a special version of the SHEF format where numbers are fixed to seven decimal points.

SSP Format

"SSP" is a binary format following the specifications of the Sutron Standard Protocol, which is useful when transmitting to XConnect or Tempest master stations.

If you are interested in knowing more about the low-level details of the protocol, please contact Sutron Customer Service.

Text Format

The Text format is an ASCII, human readable format intended to convey the current status of the sensors, quality, and alarm state. It is similar in format to the "SHOW" command, and is particularly useful for alarm messages. Sensor readings are separated by <CRLF>, while fields in the sensor data are separated with a space.

An example of a Text format message:

```
RTU01 12:22<cr><lf>
STAGE 4.55 G H+R+<cr><lf>
RAIN 2.0 G OK<cr><lf>
```

The message begins with the station's name ("RTU01") and the current time in HH:MM (12:22) format followed by a list of sensor readings containing the name of the sensor ("STAGE"), the value ("4.55"), the quality ("G"), and the alarm status ("H+R+").

In the above example "H+R+" indicates that STAGE is experiencing a high limit and high rate of change alarm, while the "OK" status for RAIN indicates that it is within expected limits.

Quality codes:

G: Good quality

B: Bad quality

U: Undefined quality

Alarm codes:

H: High limit exceeded

L: Low limit exceeded

R: High Rate of change

OK: Normal

A "+" after a code indicates that the sensor has just entered that state, while a "-" after a code indicates that the sensor has just exited that state.

BUFR

"BUFR" format is a binary message, conforming to WMO BUFR requirements for data transmissions and storage.

When BUFR is the selected format, the telemetry block used to "tag" the data for transmission has the following properties:
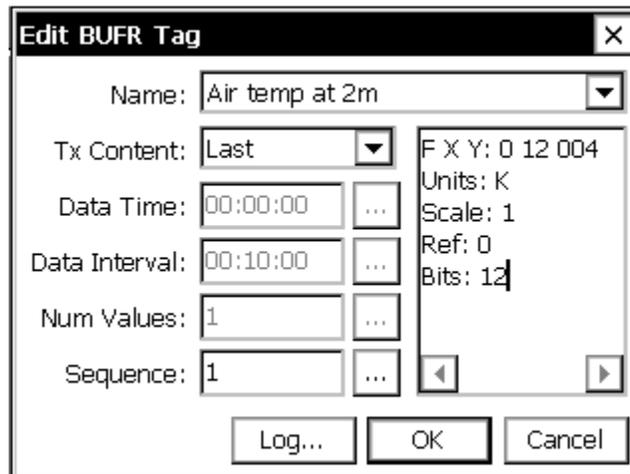
Name

The Name field describes the data to be transmitted. The Name corresponds to the Name (or sometimes called, "Description"), found in the WMO BUFR Table B. The read-only text pane to the right, updates as Name is changed, to show the following details from Table B:

Descriptor (F X Y)	The F X Y descriptor that uniquely identifies the data from Table B, e.g., 0 12 004 for "Air Temp at 2m"
Units	The units of the transmitted data
Scale	The scale applied to the data prior to transmission. Note, scale is applied as follows: transmitted data = measured value times 10 raised to the power of scale.
Ref	Reference Value subtracted from measured value after scaling, prior to transmission.
Bits	Data width in bits allowed for the transmitted value

Expanding Table B

The logger has an internal version of the WMO Table B that includes a handful of commonly transmitted values. If the internal table does not contain the values you need to transmit, create a text file called "BUFRTableB.txt", add the desired items in the format described below, and copy the text file to the logger's Flash Disk. Here is the format of the text file:

```
F,X,Y,Name,Units,Scale,Reference,Bits
```

For example:

```
0,1,1,WMO block number,Numeric,0,0,7
0,1,2,WMO station number,Numeric,0,0,10
0,5,2,Latitude (coarse accuracy),deg,2,-9000,15
0,6,2,Longitude (coarse accuracy),deg,2,-18000,16
0,7,4,Pressure,Pa,-1,0,14
0,11,1,Wind direction,degree true,0,0,9
0,11,2,Wind speed,m/s,1,0,12
0,11,35,Vertical gust acceleration,m s-2,2,-8192,14
0,12,1,Temperature/air temperature,K,1,0,12
0,12,4,Air temp at 2m,K,1,0,12
0,13,3,Relative humidity,%,0,0,7
0,13,23,Total precip past 24 hours,kg m-2,1,-1,14
0,13,55,Intensity of precipitation,kg m-2 s-1,4,0,8
0,14,30,Direct sol rad/period (high accuracy),J m-2,-2,0,20
0,14,32,Total sunshine,h,0,0,10
0,25,25,Battery voltage,V,1,0,9
```

Tx Content, Data Interval, Data Time, and Num Values

These properties define what data to include in the transmission. See the following table for details:

Tx Content		Description of data to transmit
All		All the data measured since the last transmission

Individual		Individual values are selected based on timestamp
	Data Interval	The time interval between readings. For example, "01:00:00" means "hourly data"
	Data Time	The time, or offset into the interval, of the data to include. For example, "00:15:00" means "include data measured on a 15 minute offset"
	Num Values	Number of values for each data item to include in the transmission. Enabled only for non-interleaved transmission formats (when interleaved, the Num Values setting associated with the telemetry device is used)
Last		Only the last measured value will be transmitted (when interleaved, the Num Values setting associated with the telemetry device determines number of values transmitted)
Exclude		Exclude this item from transmission

Sequence

This property determines the order of the data item within the transmission.

The "Log…" Button

The "Log…" button opens a dialog used to select the backing log containing the data for transmission.
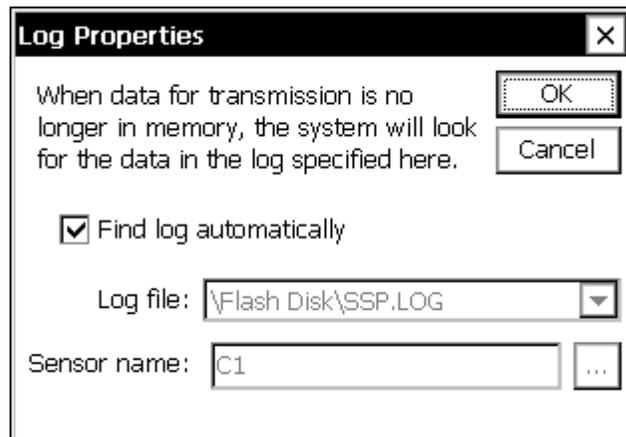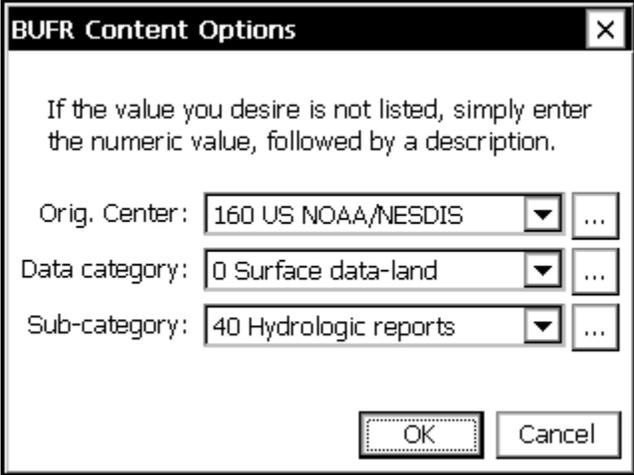

The system normally stores transmission data in memory, and only resorts to searching the log when the data is not found there (e.g., due to power-loss). The system normally selects the backing log automatically. If the automatic selection is incorrect, uncheck "Find log automatically" and specify the log manually.

Note: The system is unable to use log records as a backing log for transmission data (i.e., data logged using the LogRec and LogField blocks).

BUFR Content Options

When the selected format is "BUFR", the telemetry device settings (e.g., Satlink, LANTx, GPRS, or Iridium), offer the following BUFR content option settings for edit (accessed by pressing the "BUFR" button on the telemetry device settings dialog):

Originating Center

Select the originating to use in the BUFR message. If the desired center is not listed in the dialog drop-down, simply enter the desired value, with its numeric component, first (e.g., "34 - My originating center"). If the field is left blank, 0 is used.

Data Category

Select the data category to use in the BUFR message. If the desired category is not listed in the dialog drop-down, simply enter the desired value, with its numeric component, first. If the field is left blank, 0 is used.

Data Sub-category

Select the data subcategory to use in the BUFR message. If the desired subcategory is not listed in the dialog drop-down, simply enter the desired value, with its numeric component, first. If the field is left blank, 0 is used.

APPENDIX B – IRIDIUM TRANSMISSION HEADER

This appendix describes the content and format of the header included in all Iridium transmissions (so long as "Include Header" is selected in the setup).

The Facts

Iridium modem buffers are limited in size, forcing software to limit the length of messages it can send at any one given time. If messages are allowed to exceed these buffer lengths, then multiple transmissions must be made to send a complete message. These messages include self-timed and alarm reports, command responses, file data, and log data. The 7/8310 has the requirement that it reply with command responses over Iridium as soon as the command is processed, which, given the way the 7/8310 does message checking, may occur between two halves of a self-timed transmission. In this scenario, the receiver will receive "interleaved" partial responses.

While packets of different types may "interleave" as described, the assumption is the packets of any given message will always arrive "in order" relative to each other (and, if not, the MOMSM number w/in the Iridium Gateway header can be used to determine order).

The Issues

The above raises the following questions:

- How does the receiver identify the scope of a packet (i.e., is the packet a complete message or only a partial)?

- How does the receiver differentiate between self-timed, alarm, and command response transmissions?

- How does the receiver know how to "stitch" together "interleaved" partial responses (e.g., partial self-timed followed by command response followed by remaining self-timed)?

- How do we transfer log and file data?

The Solution

The solution is to use a header and optional sub-header to identify the content of the message such that the receiver would be able to reassemble packets. Here are the characteristics of the header and sub-header:

- The header uses a single byte in the ASCII printable range to make it easy for humans to interpret the content

- The header byte identifies the packet type in terms of the content of the packet, e.g., self-timed, self-timed extended, entering alarm, entering alarm extended, etc. The "extended" types mean multiple packets are required to transmit the entire message.

- The extended packet types include a comma-delimited sub-header to describe the subset of data being sent. The first sub-header differs from all subsequent sub-headers, in that it includes the total size of the data being sent.

- A sub-header is also included when user indicates the station name should be included in the transmission

Packet Structure:

packet-type	sub-header	data

Sub-header 0, two options:

standard:

,id,start-byte,total-bytes:

with station name:

,id,start-byte,total-bytes,station-name:

Sub-header 1 thru N:

,id,start-byte:

where,

packet-type	:=	Numeric ASCII character defining packet type. See type definitions in table, below
id	:=	Numeric ASCII text defining the message id. Starts at 0. Rolls over after 99.
total-bytes	:=	Numeric ASCII text defining the total number of data bytes to be sent (data only, does not include overhead bytes)
start-byte	:=	Numeric ASCII text defining which byte of total-bytes is the start byte of the current packet. Starts at 0.
station-name	:=	Optional ASCII field that has station name. Formatted as ,N=STATION NAME

Packet Types

Hex Value	ASCII	Description
0x30	0	Self-timed
0x31	1	Self-timed extended
0x32	2	Entering alarm

Hex Value	ASCII	Description
0x33	3	Entering alarm extended
0x34	4	Exiting alarm
0x35	5	Exiting alarm extended
0x36	6	Command response
0x37	7	Command response extended
0x38	8	Forced transmission
0x39	9	Forced transmission extended
...		Reserved for future use
0x7D	}	User defined
0x7E	~	Look to next byte for meaning
0xFF		Binary data, reserved for future use