

Figure 4.87 - Coexistence View Format Menu - Show Tooltips on Computer Screen

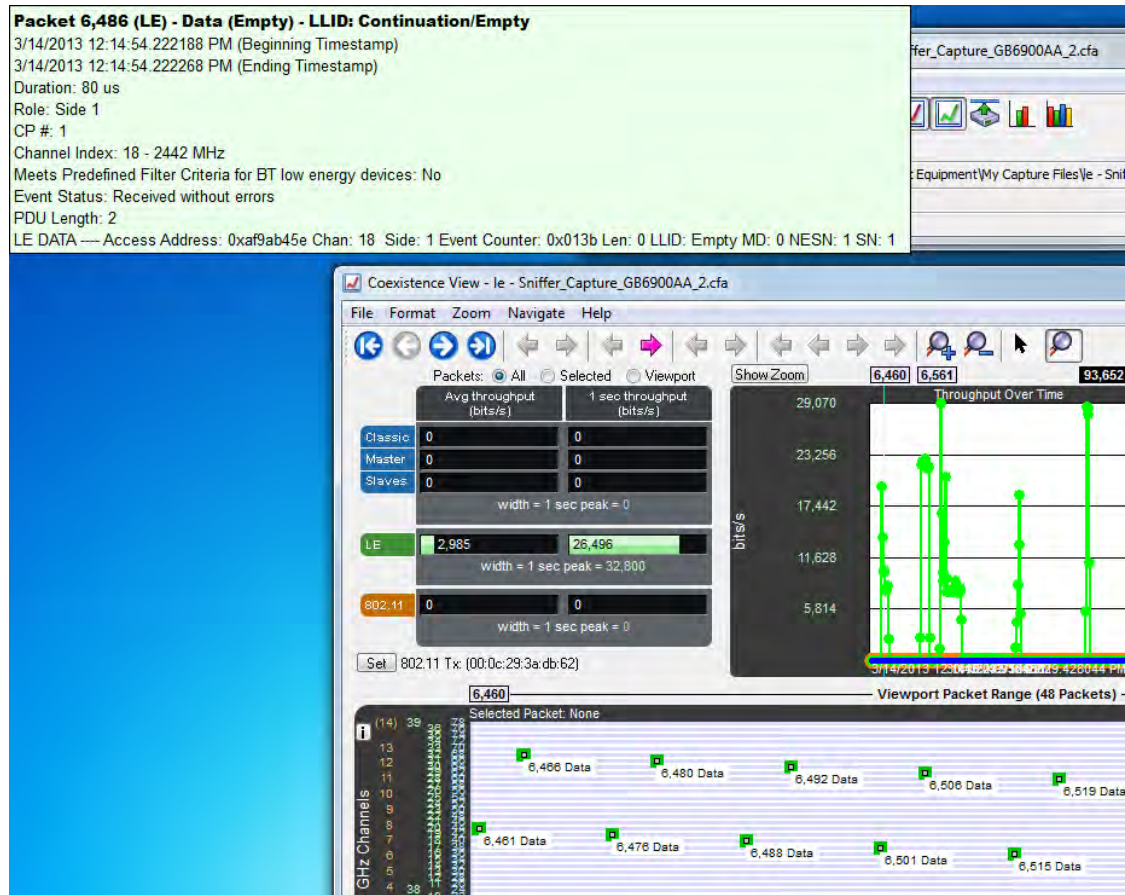


Figure 4.88 - Coexistence View Timeline Tool Tip Shown Anchored to Computer Screen

4.3.4.29 The two Timelines

There are two **Timelines** available for viewing, one for the 5 GHz range and one for the 2.4 GHz range. Classic *Bluetooth* and *Bluetooth* low energy occur only in the 2.4 GHz range. 802.11 can occur in both.

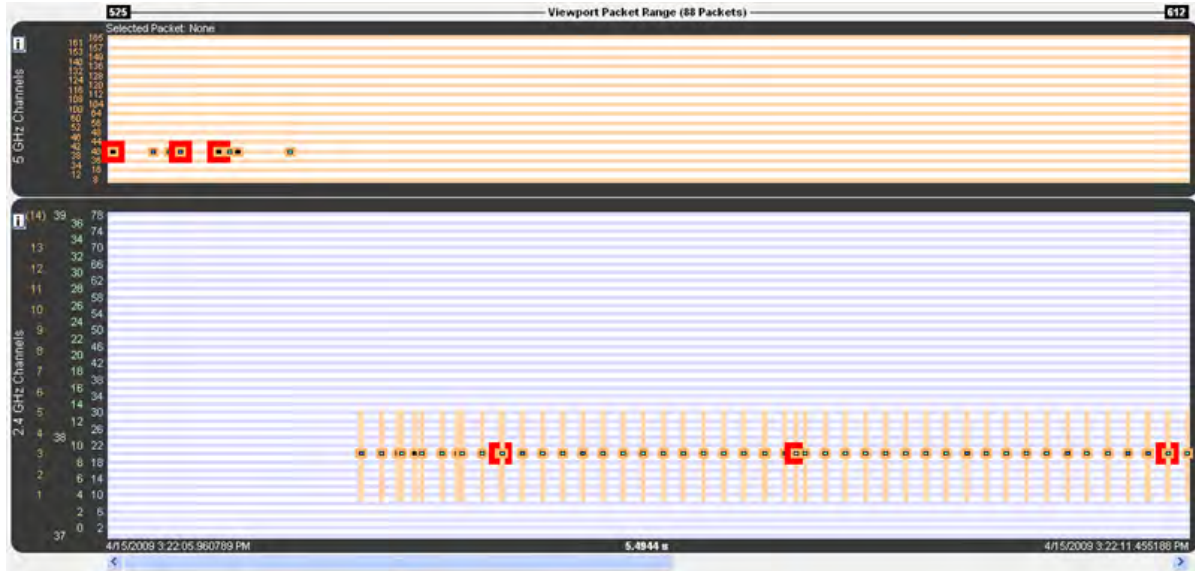


Figure 4.89 - 5 GHz and 2.4 GHz 802.11 packets

The y-axis labels show the channels for each technology and are color-coded: Blue = Classic *Bluetooth*, Green = *Bluetooth* low energy, Orange = 802.11.

The 5 GHz timeline has only 802.11 channel labels, and the rows alternate orange and white, one row per channel.

The 2.4 GHz timeline has labels for all three technologies. The rows alternate blue and white, one row per Classic *Bluetooth* channel. The labels going left-to-right are 802.11 channels, *Bluetooth* low energy advertising channels, *Bluetooth* low energy regular channels, and Classic *Bluetooth* channels.

The **Viewport Packet Range** above the timelines shows the packet range and packet count of packets that would be visible if both timelines were shown (i.e. hiding one of the timelines doesn't change the packet range or count). This packet range matches the packet range shown above the viewport in the [Throughput Graph](#), as it must since the viewport defines the time range used by the timelines. When no packets are in the time range, each of the two packet numbers is drawn with an arrow to indicate the next packet in each direction and can be clicked to navigate to that packet (the packet number changes color when the mouse pointer is placed on it in this case).

< 15,417 An arrow points to the next packet when no packets are in the time range.

< 15,417 An arrowed packet number changes color when the mouse pointer is on it. Clicking navigates to that packet.

The header shows information for packets that are selected.

The footer shows the beginning/ending timestamps and visible duration of the timelines.

The 'i' buttons bring up channel information windows, which describe channel details for each technology. They make for interesting reading.

802.11 5 GHz

Only channels with a base value of 5 GHz and spacings of either 20 or 40 MHz are shown here. Due to space limitations, each channel is drawn with fixed spacing instead of being spaced relative to its distance from other channels as is done with 2.4 GHz channels (with the exception of 802.11 channel 14).

Figure 4.90 - 5 GHz information window

Bluetooth Classic
 There are 79 Classic channels. Each channel is 1 MHz wide and has the indicated center frequency. Channels do not overlap.
 0 = 2402 MHz 10 = 2412 MHz 20 = 2422 MHz 30 = 2432 MHz 40 = 2442 MHz 50 = 2452 MHz 60 = 2462 MHz 70 = 2472 MHz
 1 = 2403 MHz 11 = 2413 MHz 21 = 2423 MHz 31 = 2433 MHz 41 = 2443 MHz 51 = 2453 MHz 61 = 2463 MHz 71 = 2473 MHz
 2 = 2404 MHz 12 = 2414 MHz 22 = 2424 MHz 32 = 2434 MHz 42 = 2444 MHz 52 = 2454 MHz 62 = 2464 MHz 72 = 2474 MHz
 3 = 2405 MHz 13 = 2415 MHz 23 = 2425 MHz 33 = 2435 MHz 43 = 2445 MHz 53 = 2455 MHz 63 = 2465 MHz 73 = 2475 MHz
 4 = 2406 MHz 14 = 2416 MHz 24 = 2426 MHz 34 = 2436 MHz 44 = 2446 MHz 54 = 2456 MHz 64 = 2466 MHz 74 = 2476 MHz
 5 = 2407 MHz 15 = 2417 MHz 25 = 2427 MHz 35 = 2437 MHz 45 = 2447 MHz 55 = 2457 MHz 65 = 2467 MHz 75 = 2477 MHz
 6 = 2408 MHz 16 = 2418 MHz 26 = 2428 MHz 36 = 2438 MHz 46 = 2448 MHz 56 = 2458 MHz 66 = 2468 MHz 76 = 2478 MHz
 7 = 2409 MHz 17 = 2419 MHz 27 = 2429 MHz 37 = 2439 MHz 47 = 2449 MHz 57 = 2459 MHz 67 = 2469 MHz 77 = 2479 MHz
 8 = 2410 MHz 18 = 2420 MHz 28 = 2430 MHz 38 = 2440 MHz 48 = 2450 MHz 58 = 2460 MHz 68 = 2470 MHz 78 = 2480 MHz
 9 = 2411 MHz 19 = 2421 MHz 29 = 2431 MHz 39 = 2441 MHz 49 = 2451 MHz 59 = 2461 MHz 69 = 2471 MHz
 The row labels are placed at the center frequency of each channel.

Bluetooth low energy (LE)
 There are 40 LE channels. Each channel is 2 MHz wide and has the indicated center frequency. Channels do not overlap.
 Channels 0 through 36 are Data channels. Channels 37 through 39 are Advertising channels.
 37 = 2402 MHz 4 = 2412 MHz 9 = 2422 MHz 13 = 2432 MHz 18 = 2442 MHz 23 = 2452 MHz 28 = 2462 MHz 33 = 2472 MHz
 0 = 2404 MHz 5 = 2414 MHz 10 = 2424 MHz 14 = 2434 MHz 19 = 2444 MHz 24 = 2454 MHz 29 = 2464 MHz 34 = 2474 MHz
 1 = 2406 MHz 6 = 2416 MHz 38 = 2426 MHz 15 = 2436 MHz 20 = 2446 MHz 25 = 2456 MHz 30 = 2466 MHz 35 = 2476 MHz
 2 = 2408 MHz 7 = 2418 MHz 11 = 2428 MHz 16 = 2438 MHz 21 = 2448 MHz 26 = 2458 MHz 31 = 2468 MHz 36 = 2478 MHz
 3 = 2410 MHz 8 = 2420 MHz 12 = 2430 MHz 17 = 2440 MHz 22 = 2450 MHz 27 = 2460 MHz 32 = 2470 MHz 39 = 2480 MHz
 The row labels are placed at the center frequency of each channel.

802.11 2.4 GHz
 In the 802.11 2.4 GHz frequency range there are 11 channels in the USA, 13 in Europe, and 14 in Japan. Each channel is 22 MHz wide. Channels overlap.
 There is a 5 MHz shift between each of the first 13 channels. There is a 12 MHz shift between channels 13 and 14.
 1 = 2401-2423 MHz (centered at 2412 MHz) (USA, Europe, Japan) 8 = 2436-2458 MHz (centered at 2447 MHz) (USA, Europe, Japan)
 2 = 2406-2428 MHz (centered at 2417 MHz) (USA, Europe, Japan) 9 = 2441-2463 MHz (centered at 2452 MHz) (USA, Europe, Japan)
 3 = 2411-2433 MHz (centered at 2422 MHz) (USA, Europe, Japan) 10 = 2446-2468 MHz (centered at 2457 MHz) (USA, Europe, Japan)
 4 = 2416-2438 MHz (centered at 2427 MHz) (USA, Europe, Japan) 11 = 2451-2473 MHz (centered at 2462 MHz) (USA, Europe, Japan)
 5 = 2421-2443 MHz (centered at 2432 MHz) (USA, Europe, Japan) 12 = 2456-2478 MHz (centered at 2467 MHz) (Europe, Japan)
 6 = 2426-2448 MHz (centered at 2437 MHz) (USA, Europe, Japan) 13 = 2461-2483 MHz (centered at 2472 MHz) (Europe, Japan)
 7 = 2431-2453 MHz (centered at 2442 MHz) (USA, Europe, Japan) 14 = 2473-2495 MHz (centered at 2484 MHz) (Japan)
 The row labels for 802.11 channels 1-13 are placed at the center frequency of each channel.
 The row label for 802.11 channel 14 is in parentheses because that channel's center frequency is above the top of the graph.

Figure 4.91 - 2.4 GHz information windows

4.3.4.30 Bluetooth slot markers

When zoomed in far enough *Bluetooth* slot markers appear in the 2.4 GHz timeline. A *Bluetooth* slot is 625 μs wide.

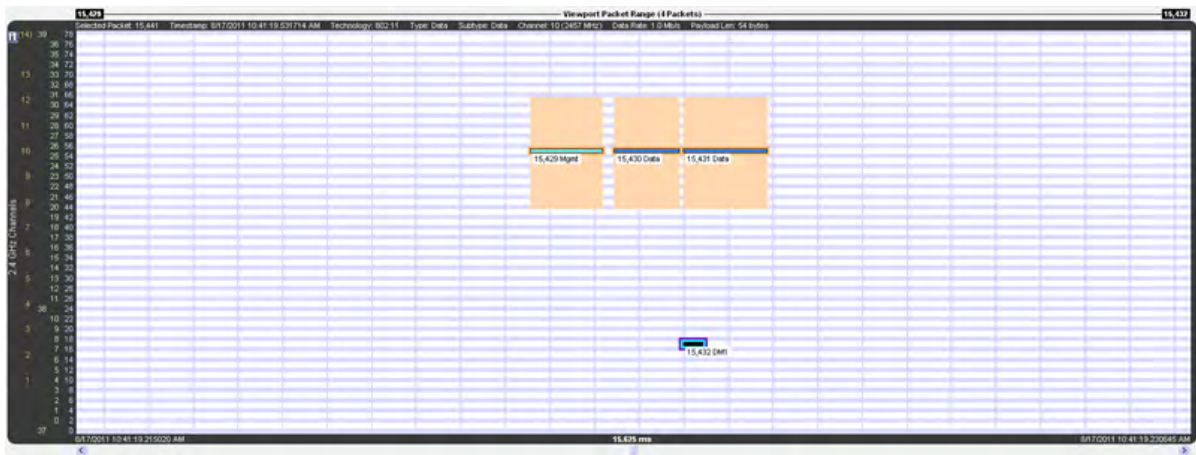


Figure 4.92 - Vertical blue lines are *Bluetooth* slot markers

4.3.4.31 Zooming

There are various ways to zoom:

1. Drag one of the sides of the **Throughput Graph** viewport.
2. Select a zoom preset from the **Zoom** or right-click menus.
3. Select the **Zoom In** or **Zoom Out** button or menu item.
4. Turn the mouse wheel in the **Timelines** or the **Zoomed Throughput Graph** while the zoom cursor is selected. The action is the same as selecting the **Zoom In** and **Zoom Out** buttons and menu items except that the time point at the mouse pointer is kept in place if possible.
5. Select the **Zoom to Data Point Packet Range** menu item, which zooms to the packet range shown in the most recently displayed tool tip.
6. Select the **Zoom to Selected Packet Range** menu item, which zooms to the selected packet range as indicated in the **Selected Packets** text in the timeline header.
7. Select the **Custom Zoom** menu item. This is the zoom level from the most recent drag of a viewport side, selection of **Zoom to Data Point Packet Range**, or selection of **Zoom to Selected Packet**.

The zoom buttons and tools step through the zoom presets and custom zoom, where the custom zoom is logically inserted in value order into the zoom preset list for this purpose.

4.3.4.32 Discontinuities

([Click here to see a Timeline Discontinuities video...](#))

A discontinuity is when the timestamp going from one packet to the next either goes backward by any amount or forward by more than 4.01 s (this value is used because the largest possible connection interval in *Bluetooth* low energy is 4.0 s). A discontinuity is drawn as a vertical cross-hatched area one *Bluetooth* slot (625 μ s) in width. A discontinuity for a timestamp going backward is called a negative discontinuity and is shown in red. A discontinuity for a timestamp going forward by more than 4.01 s is called a positive discontinuity and is shown in black. A positive discontinuity is a cosmetic nicety to avoid lots of empty space. A negative discontinuity is an error.

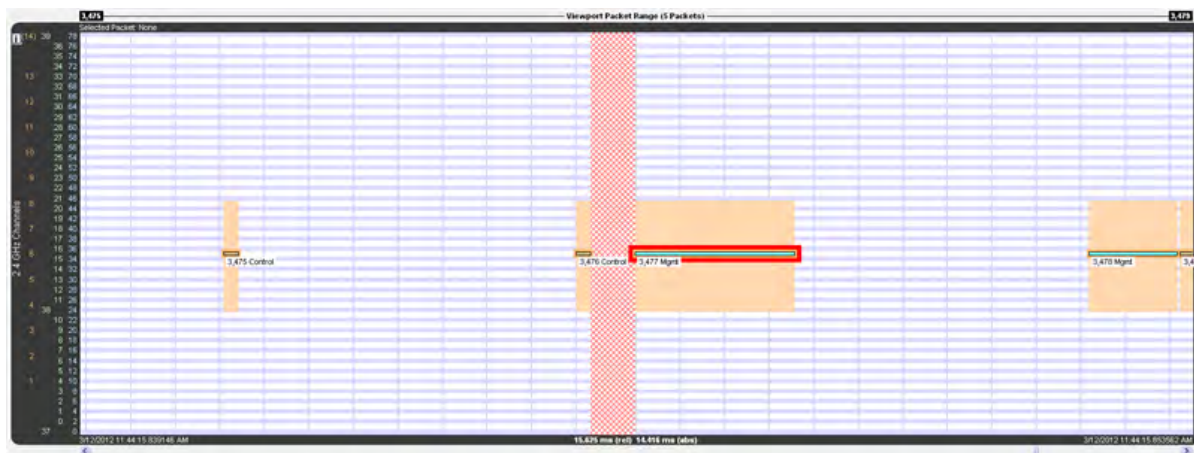


Figure 4.93 - A negative discontinuity

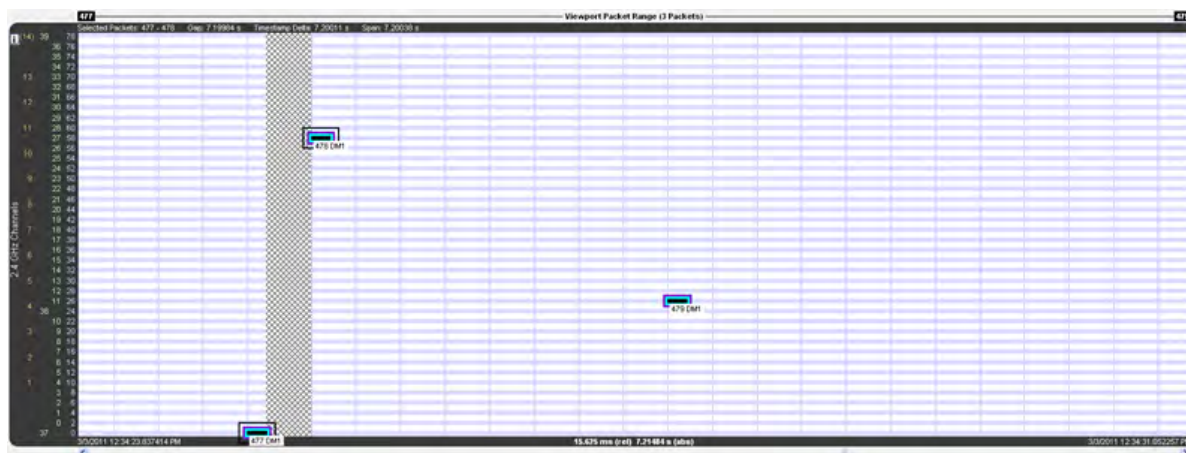


Figure 4.94 - A positive discontinuity

When there are one or more discontinuities the actual time encompassed by the visible timeline differs from the zoom level duration that would apply in the absence of any discontinuities. The actual time, referred to as absolute time, is shown followed by “(abs)”. The zoom level duration, referred to as relative time, is shown followed by “(rel)”. When there are no discontinuities, relative and absolute time are the same and a single value is shown.

Selected Packets: 477 - 478 Gap: 7.19984 s Timestamp Delta: 7.20011 s Span: 7.20038 s

Figure 4.95 - Timeline header with discontinuity

15.625 ms (rel) 7.21484 s (abs)

Figure 4.96 - Timeline duration footer with discontinuity

For example, the timeline above has a zoom level duration of 15.625 ms (the relative time shown in the footer). But the discontinuity graphic consumes the width of a *Bluetooth* slot (625 μs), and that area is 7.19984 s of absolute time as shown by the Gap value in the header. So the absolute time is 7.21484 s:

Zoom level duration – *Bluetooth* slot duration + Gap duration =

$$15.625 \text{ ms} - 625 \mu\text{s} + 7.19984 \text{ s} =$$

$$0.015625 \text{ s} - 0.000625 \text{ s} + 7.199840 \text{ s} =$$

$$0.015000 \text{ s} + 7.199840 \text{ s} =$$

$$7.214840 \text{ s} =$$

$$7.21484 \text{ s}$$

4.3.4.33 High-Speed *Bluetooth*

High-speed *Bluetooth* packets, where *Bluetooth* content hitches a ride on 802.11 packets, have a blue frequency range box instead of orange as with regular 802.11 packets (both are shown below), and the tool tip has two colors, orange for 802.11 layers and blue for *Bluetooth* layers.



Figure 4.97 - High-speed Bluetooth packets have a blue frequency box and a two-tone tool tip

4.3.4.34 Coexistence View - No Packets Displayed with Missing Channel Numbers

Note: This topic applies only to Classic Bluetooth.

Captured packets that don't contain a channel number, such as HCI and BTSnoop, will not be displayed. When no packets have a channel number the **Coexistence View Throughput Graph** and **Timelines** will display a message: "Packets without a channel number (such as HCI) won't be shown."

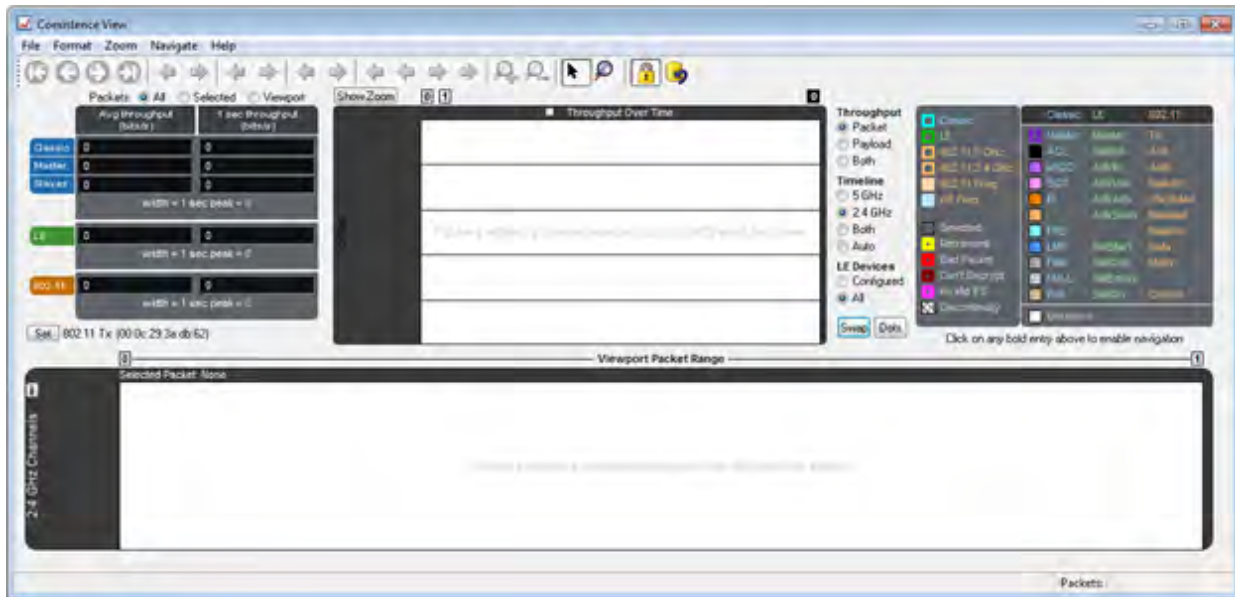


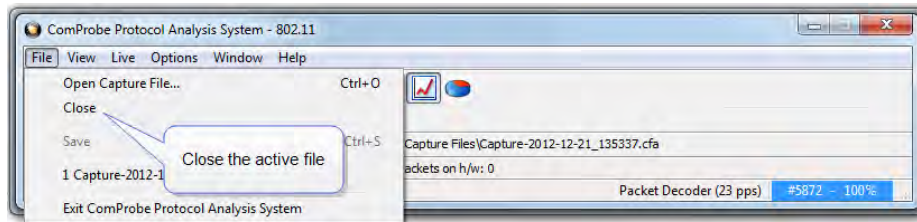
Figure 4.98 - Missing Channel Numbers Message in Timelines

4.3.4.35 High Speed Live View

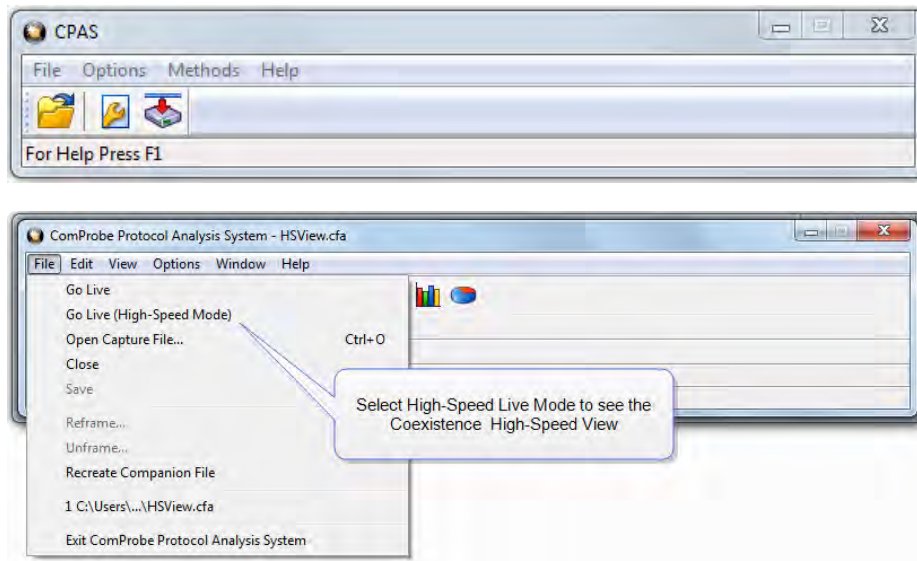
When using the Frontline® 802.11 in conjunction with other ComProbe devices, or in a stand-alone configuration, a smaller version of the standard **Coexistence View** is available. This **High Speed Live View** is essentially the **Viewport** from the standard **Coexistence View**.



When viewing **High Speed Live**, only 802.11 traffic is visible. Because *Bluetooth* packets are slow they are not visible in High Speed mode.

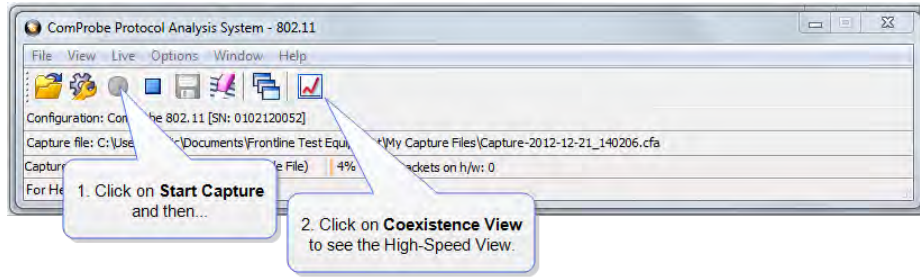
1. Click on the **Control** window **File** menu and select **Close**.



2. The **Control** window will open again. Click on the Control Window **File** menu and select **Go Live (High-Speed Mode)**



3. Click on the **Control** window **Start Capture** button  to begin capturing data. Click on the **Coexistence View** button  and the **High-Speed View** will appear.



The Coexistence View (High Speed Live Mode) window will appear.

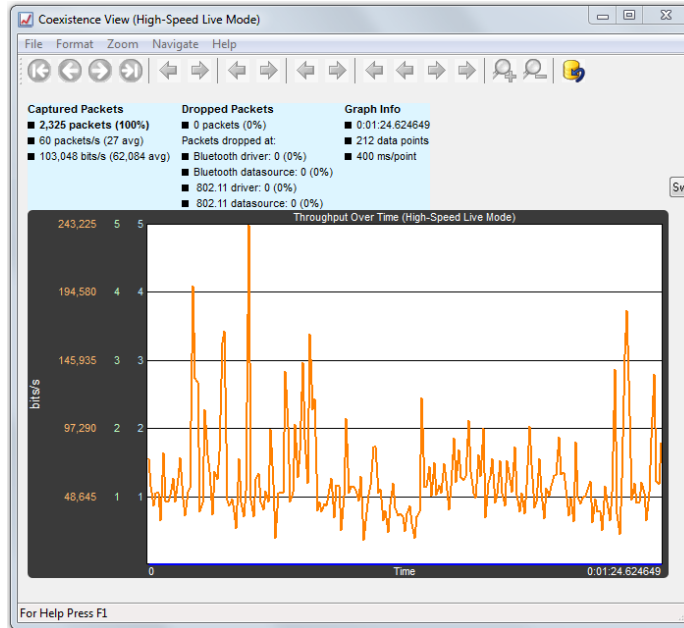


Figure 4.99 - High-Speed Live Window

4.3.5 Message Sequence Chart (MSC)

The **Message Sequence Chart** (MSC) displays information about the messages passed between protocol layers. MSC displays a concise overview of a *Bluetooth* connection, highlighting the essential elements for the connection. At a glance, you can see the flow of the data including role switches, connection requests, and errors. You can look at all the packets in the capture, or filter by protocol or profile. The MSC is color coded for a clear and easy view of your data.

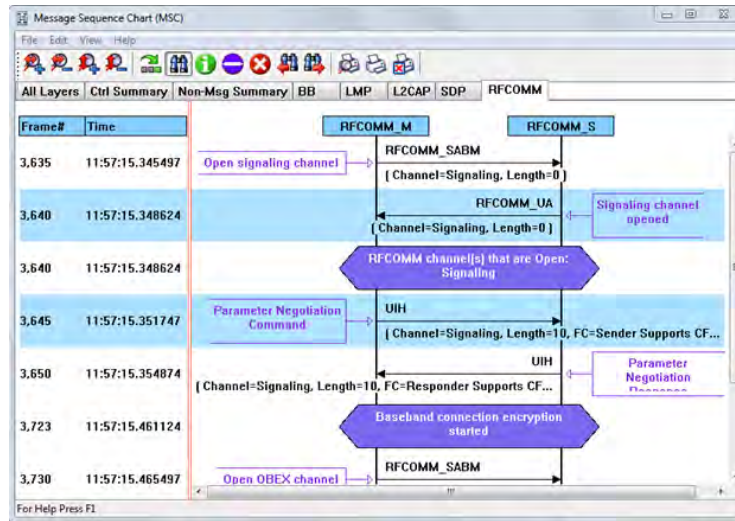


Figure 4.100 - Message Sequence Chart Window

How do I access the chart?

You access the **Message Sequence Chart** by selecting the icon or **MSC Chart** from the **View** menu from the **Control** window or **Frame Display**.

What do I see on the dialog?



At the top of the dialog you see four icons that you use to zoom in and out of the display vertically and horizontally. The same controls are available under the **View** menu.

There are three navigation icons also on the toolbar.

| | |
|--|--|
| | This takes you to the first Information Frame. |
| | This takes you to first Protocol State Message. |
| | This takes you to the first Error Frame. Click here to learn more about this option. |

If there is both Classic and low energy packets, there will be a **Classic** and **LE** tab at the top of the dialog.

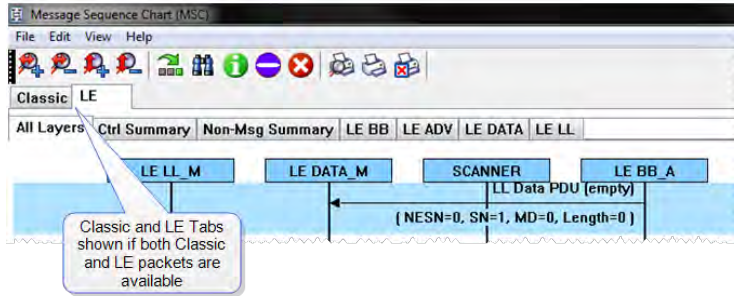
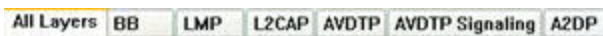


Figure 4.101 - Classic and LE tabs

If the **Classic** tab is selected, you will see Classic protocols. If you select the **LE** tab, you will see LE Protocols. If there is only Classic or only LE, the Classic and LE tabs will not appear.



Also along the top of the dialog are a series of protocol tabs. The tabs will vary depending on the captured protocols.

Clicking on a tab displays the messaging between the master and slave for that protocol. For example, if you select **RFCOMM**, you will see the messaging between the **RFCOMM{M}** Master, and the **RFCOMM{S}** Slave.



The Non-Message Summary tab displays all the non-message items in the data.

The **Ctrl Summary** tab displays the signaling packets for all layers in one window in the order in which they are received.

The information in the colored boxes displays general information about the messaging. The same is true for each one of the protocols.

If you want to see the all the messaging in one dialog, you select the **All Layers** tab.

When you move the mouse over the message description you see an expanded tool tip.



If you position the cursor outside of the message box, the tool tip will only display for a few seconds.

If, however, you position the cursor within the tool tip box, the message will remain until you move the cursor out of the box.

Additionally, If you right click on a message description, you will see the select Show all Layers button.

When you select **Show all Layers**, the chart will display all the messaging layers.

The **Frame#** and **Time** of the packets are displayed on the left side of the chart.

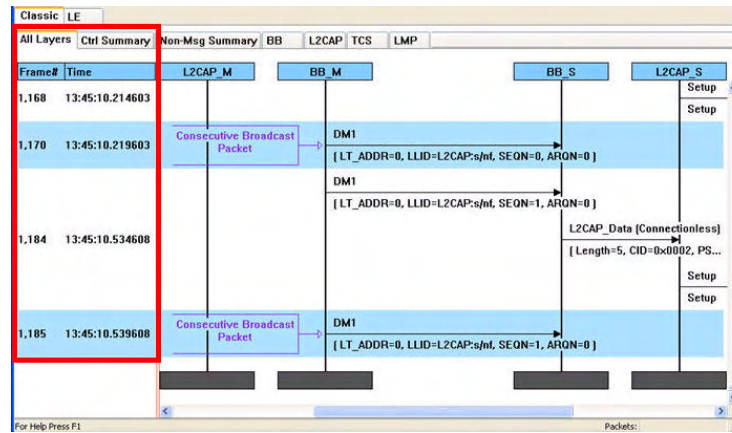


Figure 4.102 - Frame# and Time Display, inside red box.

If you click on the description of the message interaction, the corresponding information is highlighted in [Frame Display](#).

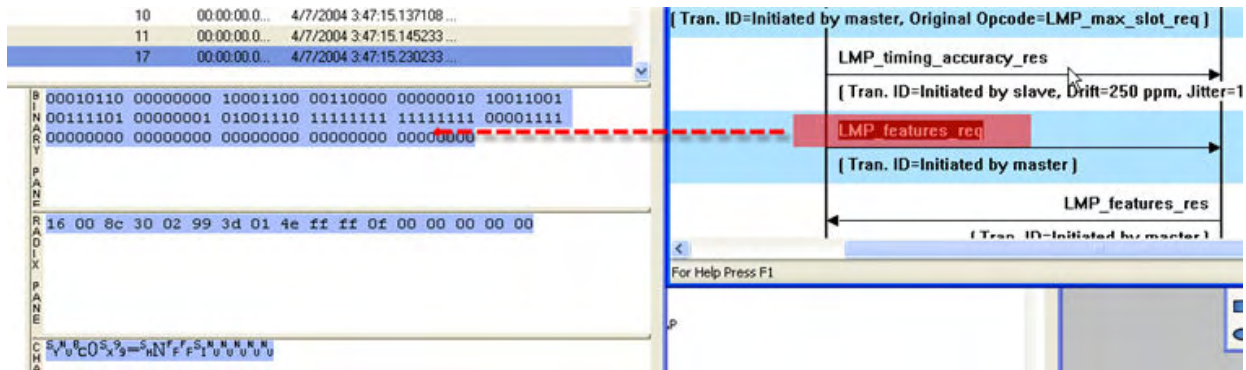


Figure 4.103 - MSC Synchronization with Frame Display

How do I navigate in the dialog?

You can use the navigation arrows at the bottom and the right side of the dialog to move vertically and horizontally. You can also click and hold while moving the pointer within dialog that brings up a directional arrow that you can use to move left/right and up/down.

Ctrl Summary tab

When you select the **Ctrl Summary tab** you will see a summary of the control and signaling frames in the order that they are received/transmitted from and to devices.

| Frame# | Role | BD_ADDR | LT_ADDR | Message | Parameter |
|---------|------|---------|---------|-------------------------|-----------------|
| 107,238 | S | | 1 | AVDTP_SUSPEND | |
| 107,240 | S | | 1 | LMP_accepted | |
| 107,242 | M | | 1 | LMP_max_slot_req | |
| 107,250 | S | | 1 | LMP_accepted | |
| 107,384 | S | | 1 | LMP_preferred_rate | |
| 109,014 | S | | 1 | LMP_sniff_req | Sniff request |
| 109,018 | M | | 1 | LMP_accepted | |
| 110,388 | S | | 1 | LMP_preferred_rate | |
| 110,560 | M | | 1 | LMP_unsniff_req | UnSniff request |
| 110,563 | S | | 1 | LMP_accepted | |
| 110,567 | M | | 1 | LMP_remove_SCO_link_req | Remove SCO link |
| 110,569 | S | | 1 | LMP_accepted | |
| 110,570 | M | | 1 | LMP_max_slot_req | |
| 110,571 | M | | 1 | LMP_max_slot_req | |
| 110,572 | S | | 1 | LMP_accepted | |
| 110,573 | S | | 1 | LMP_sniff_req | Sniff request |
| 110,574 | M | | 1 | LMP_accepted | |

Figure 4.104 - Control and Signaling Frames Summary

The frame numbered is shown, whether the message comes from the Master or Slave, the message Address, the message itself, and the timestamp.

Additionally, the control/signaling packets for each layer are shown in a different background color.

| Frame# | Role | BD_ADDR | LT_ADDR | Message | Parameter |
|--------|------|--------------|---------|--------------------|-----------|
| 85 | M | 000272b00c0e | 1 | RFCOMM_SABM | Signaling |
| 87 | M | 000272b00c0e | 1 | LMP_preferred_rate | |
| 89 | S | | 1 | LMP_preferred_rate | |
| 91 | S | | 1 | RFCOMM_UA | |
| 97 | M | 000272b00c0e | 1 | RFCOMM_SABM | OBEX |
| 99 | S | | 1 | RFCOMM_UA | |
| 109 | M | 000272b00c0e | 1 | OBEX_Connect | BIP |
| 111 | S | | 1 | OBEX_Success | |
| 113 | M | 000272b00c0e | 1 | LMP_decr_power_req | |

Figure 4.105 - Packet Layers Shown in Different Colors

If you right click within the **Ctrl Summary**, you can select **Show in MSC**.

| Frame# | Role | BD_ADDR | LT_ADDR | Message | Parameter |
|---------|------|---------|---------|--------------------|---------------|
| 107,238 | S | | 1 | AVDTP_SUSPEND | |
| 107,240 | S | | 1 | LMP_accepted | |
| 107,242 | M | | 1 | LMP_max_slot_req | |
| 107,250 | S | | 1 | LMP_accepted | |
| 107,384 | S | | 1 | LMP_preferred_rate | |
| 109,014 | S | | 1 | LMP_sniff_req | Sniff request |
| 109,018 | M | | 1 | LMP_accepted | |

Figure 4.106 - Right-Click in Ctrl Summary to Display Show in MSC

The window then displays the same information, but in the normal MSC view.

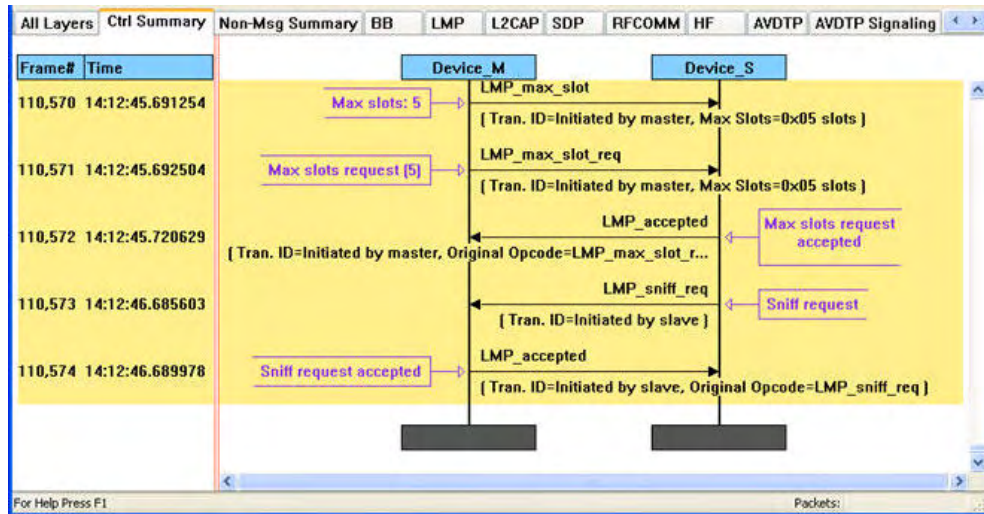


Figure 4.107 - MSC View of Selected Packet from Ctrl Summary

You can return to the text version by using a right click and selecting **Show in Text**.

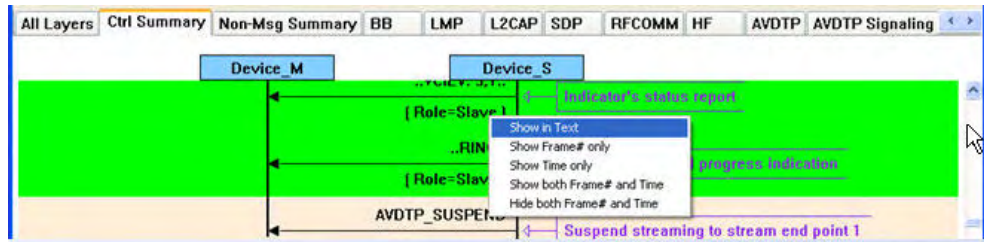


Figure 4.108 - Return to Text View Using Right-Click Menu

You can also choose to show:

- Frame # only
- Time only
- Show both Frame# and Time
- Hide both Frame# and Time

4.3.5.1 Message Sequence Chart Toolbar



Figure 4.109 - Message Sequence Chart Toolbar

Table 4.14 - Message Sequence Chart Tools



















| Tool | Keyboard | Description |
|---|----------|--|
|  | Ctrl + H | Zoom in horizontal - expands the chart horizontal view |

Table 4.14 - Message Sequence Chart Tools (continued)

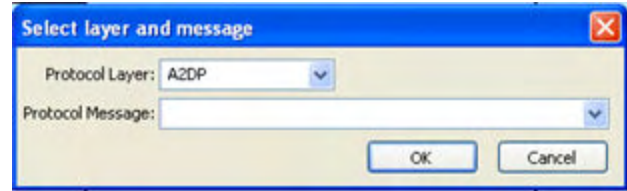
| Tool | Keyboard | Description |
|---|-----------|---|
|  | Shift + H | Zoom out horizontal - compresses the chart horizontal view |
|  | Ctrl + V | Zoom in vertical - expands the chart vertical view |
|  | Shift + V | Zoom out vertical - compresses the chart vertical view |
|  | Shift + F | Go to frame |
|  | F3 | Search |
|  | F2 | Search for prior Search  criteria. |
|  | F4 | search for Next  criteria. |
|  | Ctrl + I | Go to first information message |
|  | Ctrl + S | Go to first protocol state message |
|  | Ctrl + E | Go to first error frame |
|  | Shift + L | Lock / unlock the chart display. Clicking on the active icon or typing the keyboard command will toggle to the other state. |
|  | Ctrl + W | Print display preview |
|  | Ctrl + P | Print the display |
|  | Ctrl + C | Cancel an in-process print |

4.3.5.2 Message Sequence Chart - Search

The Message Sequence Chart has a Search function that makes it easy to find a specific type message within the layers.

When you select the 1) **Search** icon  or 2) use **F3** key, the **Select layer and message** dialog appears.

From this dialog you can search for specific protocol messages or search for the first error frame.



1. On the MSC dialog select one of the protocol tabs at the top.

Note: If you select **All Layers** in Step 1, the Protocol Layers drop-down list is active. If you select any of the other single protocols, the Protocol Layers drop-down is grayed out.

2. Or Open the Search dialog using the Search icon or the **F3** key.
3. Select a specific Protocol Message from the drop-down list.
4. Once you select the Protocol Message, click **OK**

The Search dialog disappears and the first search result is highlight in the Message Sequence Chart.

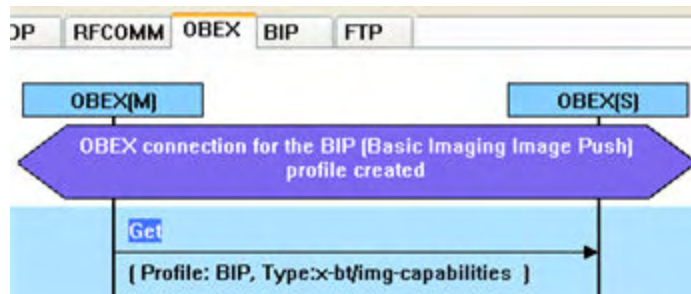
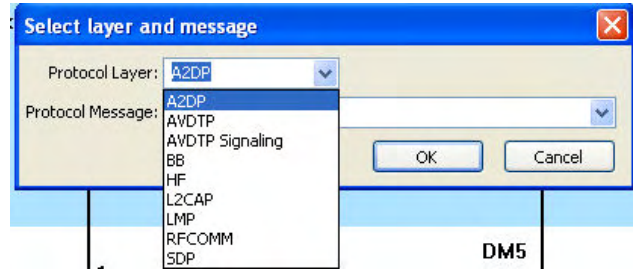


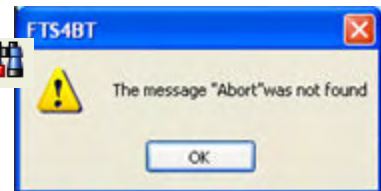


Figure 4.110 - Highlighted First Search Result


If there is no instance of the search value, you see this following dialog.


Once you have set the search value, you can 1) use the **Search Previous**  and **Search Next**  buttons or 2) **F2** and **F4** to move to the next or previous frame in the chart.

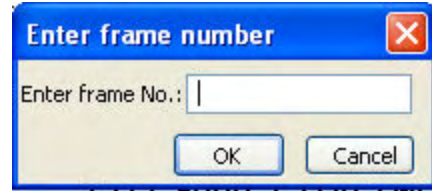


4.3.5.3 Message Sequence Chart - Go To Frame



The **Message Sequence Chart** has a **Go To Frame** function that makes it easy to find a specific frame within the layers.

In addition to [Search](#), you can also locate specific frames by clicking on the **Go To Frame**  toolbar icon.


1. Click **Go To Frame**  in the toolbar.
2. Enter a frame number in the **Enter frame No.:** text box.
3. Click **OK**.

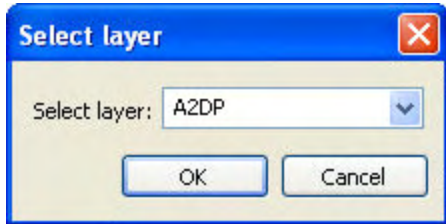


The Go To Frame dialog disappears and the selected frame is highlighted in the chart.

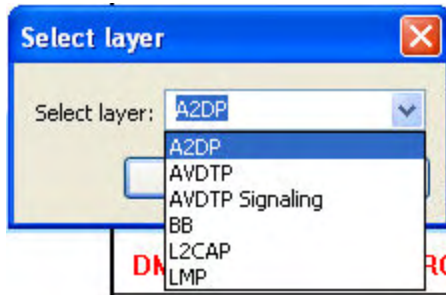
Once you have identified the frame in Go To, you can 1) use the Search Previous  and Search Next  buttons or 2) **F2** and **F4** keys to move to the next or previous frame in the chart.

4.3.5.4 Message Sequence Chart - First Error Frame

When you select **Go to first error frame** from the toolbar , the **Select layer** dialog appears.



You have to select a layer from the drop down list to choose what layer you want to search for the error.



Once you select a layer, then **OK**, the first error for that layer will be displayed.

If no error is found, a dialog will announce that event.




4.3.5.5 Message Sequence Chart - Printing



There are three standard MSC print buttons. **Print Preview**, **Print**, and **Cancel Printing**.

Print Preview

1. When you select **Print Preview** , the **Print Setup** dialog appears.
2. You next need to select your printer from the drop-down list, set printer properties, and format the print output..
3. Then you select **OK**.

After you select **OK**, the **Message Sequence Chart Print Preview** dialog appears.

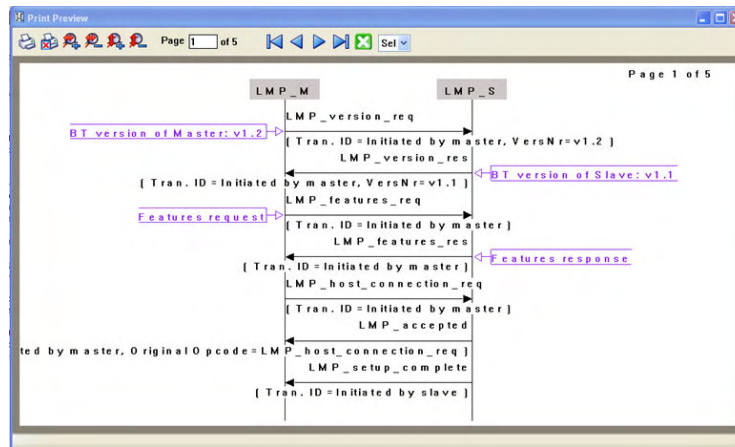


Figure 4.111 - Message Sequence Chart Print Preview

The information in the dialog will vary depending on the layer that is selected in the [Message Sequence Chart](#), the properties of the printer you select, and the amount of data in the layer (which will correspond to the number of pages displayed).

You control what you see and when to print using the toolbar at the top of the dialog.

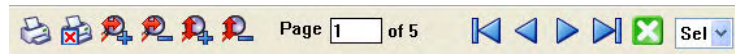


Figure 4.112 - Print Preview Toolbar

Table 4.15 - Print Preview Icons




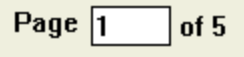



| Icon | Name | Description |
|---|-----------------|---|
|  | Print | Prints all the pages to the printer you select in Print Setup dialog. When you select Print, you will output the data that is currently being displayed. |
|  | Cancel Printing | Cancels the current printing. |

Table 4.15 - Print Preview Icons (Continued)



| Icon | Name | Description |
|---|-----------------------|--|
|  | Zoom In Horizontally | Expands the data horizontally so it can be easier to read. |
| | Zoom Out Horizontally | Squeezes the data together so that more fits on one page. |
| | Zoom In Vertically | Expands the data vertically so it can be easier to read. |
| | Zoom Out Vertically | Squeezes the data so that more fits on one page. |
|  | Current Page | <p>The current page text box displays the page number this is currently shown in the dialog.</p> <p>You can enter a number in the text box, then press Enter, and the dialog will display the data for that page.</p> |
|  | Page navigation | <p>If the data requires multiple pages, the navigation buttons will take you to:</p> <ul style="list-style-type: none"> • The first page • The previous page • The next page • The last page |
|  | Close Print Preview | Closes the dialog and returns to the Message Sequence Chart |
|  | Select Font Size | Allows selection of the print font size from the drop-down control. |

4.4 Packet Error Rate Statistics

The **Packet Error Rate** (PER) Stats view provides a dynamic graphical representation of the Packet Error Rate for each channel. The dialog displays a graph for each Classic *Bluetooth* channel numbered 0 through 78 and for each *Bluetooth* low energy channel numbered 0 through 39.

Packet Error Rate Stats assist in detecting bad communication connections. When a high percentage of re-transmits, and/or header/payload errors occur, careful analysis of the statistics indicate whether the two devices under test are experiencing trouble communicating, or the packet sniffer is having difficulty listening.

Generally, if the statistics display either a large number of re-transmits with few errors or an equal number of errors and re-transmits, then the two devices are not communicating clearly. However, if the statistics display a large number of errors and a small number of re-transmits, then the packet sniffer is not receiving the transmissions clearly.

You can access this window in Classic *Bluetooth* by selecting the **Classic Bluetooth Packet Error Rates Statistics** icon  from the **Control** window or **Frame Display**. You can access this window in *Bluetooth* low energy by selecting the **Bluetooth low energy Packet Error Rates Statistics** icon  from the **Control** window or **Frame Display**. You can also open the window from the View menu on the same windows.

Classic Bluetooth Packet Error Rate



Figure 4.113 - Classic Bluetooth PER Stats Window

Bluetooth low energy Packet Error Rate



Figure 4.114 - Bluetooth low energy PER Stats Window

4.4.1 Packet Error Rate - Channels (Classic and low energy)

The main portion of the PER Stats dialog displays the 79 individual channels, 0-78, for Classic Bluetooth® 40 individual channels, 0-39, for Bluetooth low energy.

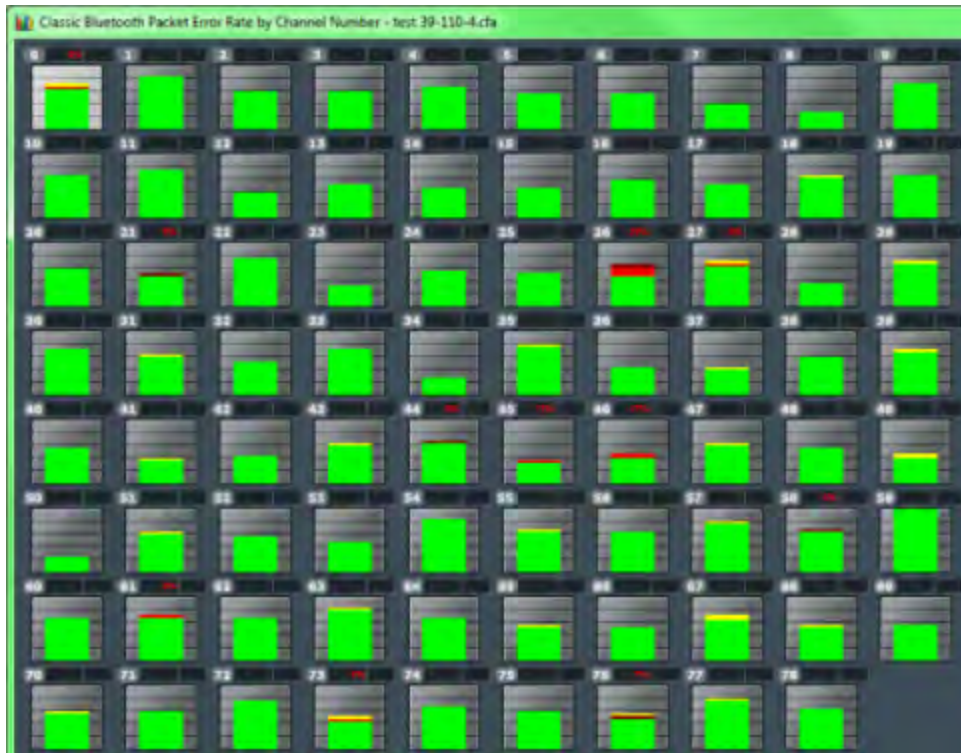


Figure 4.115 - Classic *Bluetooth* Packet Error Rate Channels

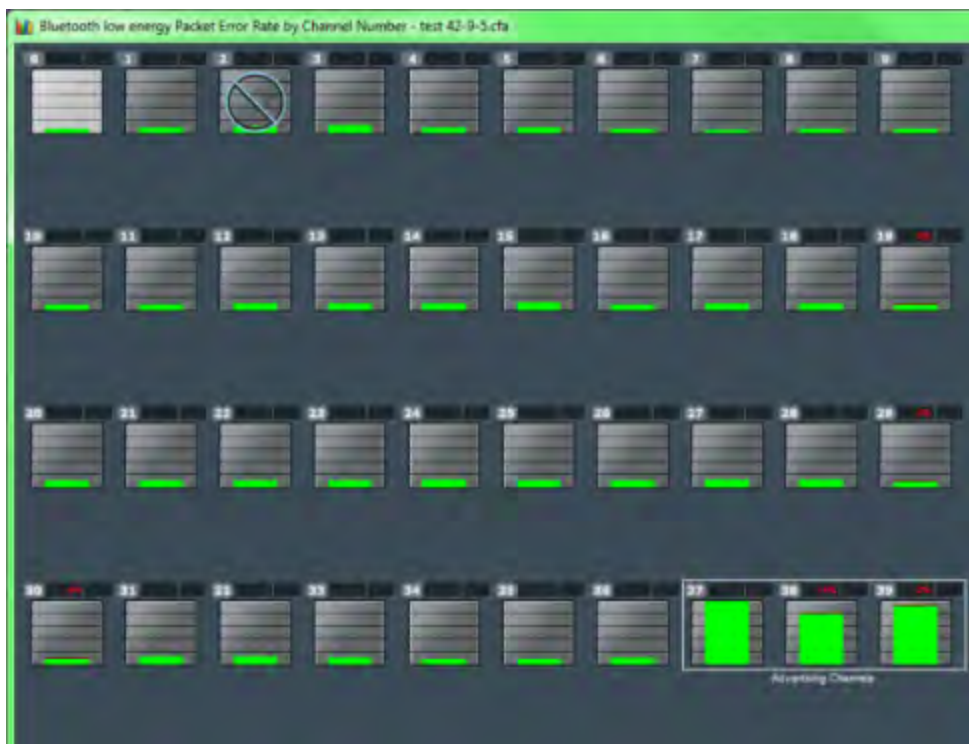


Figure 4.116 - *Bluetooth* low energy Packet Error Rate Channels

- **For Classic Bluetooth:** Each channel contains a bar that displays the number of packets with no errors in green, packets with Header Errors in red, packets with Payload or CRC errors in dark red, and Retransmitted packets in yellow.
- The red number at the top of the channel shows the percentage of Header Error and Payload/CRC Errors in relationship to the total number of packets in the channel.
- The light blue number at the top of each channel shows the [megahertz \(MHz\) for the channel if the option is chosen in the Additional Statistics section.](#)
- When you select a channel, detailed information for that channel is displayed in the [expanded chart on the upper right.](#)
- The channels change dynamically as the Viewport is moved or new data appears within the [Viewport.](#)
- The **Channel Not Available** symbol is displayed if the channel is not available in the most recent channel map that is in or before the last selected packet, even if that channel map comes before the first selected packet. *Bluetooth* Adaptive Frequency Hopping processes will block channels determined to be unreliable. These channels are not available because the Bluetooth devices have decided not to use them.
- "s" changes the size of the entire dialog.
- "c" changes the contrast of the dialog
- The **Reset** button is only available in live mode. The button will appear in the lower right-hand corner of the Channels section. Clicking on the **Reset** button will clear all prior data from PER Stats.



4.4.2 Packet Error Rate - Pie Chart and Expanded Chart



The **Expanded PER Stats Chart** (in the upper right) displays detailed information about the channel selected from the main channel dialog.

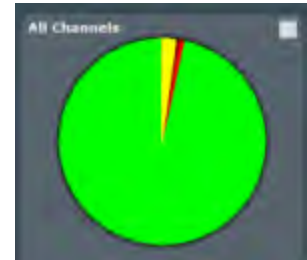
Expanded Chart



Pie Chart



- When PER Stats is first opened, Channel 0 is displayed in the expanded chart.
- The top orange number on the Y-Axis displays the maximum number of packets in Snap Mode. If Snap Mode is turned off, the number will display in light blue. For information about Snap Mode, see [Packet Error Rate - Additional Statistics on the next page](#)
- The number of the selected channel is displayed in the upper-left corner of the expanded chart.
- The combined value of Header and Payload/CRC errors for the channel is displayed in red as a percentage to the right of the channel number.
- The megahertz (MHz) value is displayed in light blue text if the MHz option is selected in the Additional Statistics section.
- The number of packets with no errors is displayed in light green in the bar chart.
- **For Classic Bluetooth®** : The number of packets that have header errors is displayed in red in the bar chart.
- **For Classic Bluetooth:** The number of payload errors is displayed in dark red in the bar chart.
- **For Classic Bluetooth:** The number of re-transmits is displayed in yellow in the bar chart.
- All the values, except MHz, change dynamically when multiple time periods are selected in the [Packet Error Rate - Scroll Bar on page 212](#).
- When you select the  in the upper-right corner, the bar chart is replaced by a pie chart. The pie chart applies to all channels, not a selected channel. To return to the bar chart, click on the channel again or click on the  in the upper right hand corner.



4.4.3 Packet Error Rate - Legend

The **Legend** displays color coded information about the channel selected.



For Classic Bluetooth

- The number of Packets with **No Errors** and percentage of packets with **No Errors** in relationship to total packets for the channel is displayed in green.
- The number of Packets with **Header Errors** and percentage of packets with **Header Errors** in relationship to total packets for the channel is displayed in red.

- The number of Packets with **Payload/CRC Errors** and percentage of packets with **Payload/CRC Errors** in relationship to total packets for the channel is displayed in dark red.
- The number of **Retransmitted** Packets and percentage of **Retransmitted** packets in relationship to total packets for the channel is displayed in yellow.
- **Total** packets and **Total** percentage is displayed in light blue.

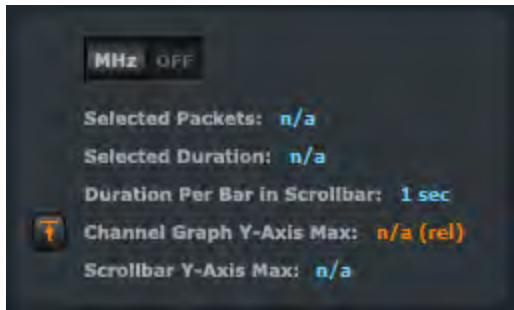
For Bluetooth low energy:

- The number of Packets with **No Errors** and percentage of packets with **No Errors** in relationship to total packets for the channel is displayed in green.
- The number of Packets with **CRC Errors** and percentage of packets with **CRC Errors** in relationship to total packets for the channel is displayed in dark red.
- **Total** packets and **Total** percentage is displayed in light blue.



For a description of the **Channel Not Available** symbol, see [PER Stats Channel](#).

4.4.4 Packet Error Rate - Additional Statistics




This Additional Statistics section of PER Stats displays information about selected packets, duration, and Y-Axis max, and it also has two controls.

- Selecting **MHz On** displays the megahertz value for each channel in the [main channels chart](#) and also in the [expanded chart](#).
- Selecting **MHz Off** removes the megahertz value.
- **Selected Packets** displays the packet range selected in the [Scroll Bar](#). This includes inapplicable packets.

Inapplicable packets include Wi-Fi packets, Sniffer Debug packets, any packets that are not relevant to PER Stats. Inapplicable packets do not appear as part of the Additional Statistics. packets.

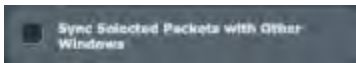
- **Selected Duration** identifies the total amount of time in the selected packet range displayed in the [Scroll Bar](#).
- **Duration Per Bar in Scrollbar:** identifies the amount of time represented by each bar in the [Scroll Bar](#).
- The **Channel Graph Y-Axis Max** can display two different values. When the **Snap Arrow** is orange , the [values for channels in the main chart](#) are shown in relative terms in **Snap Mode**. This means that one channel (or channels) with the greatest value is "snapped" to the top of the chart. In the graphic below left, Channel 33 is snapped to the top of the chart.



The channel(s) with the greatest value become a full-scale reference display for the other channels that have been relatively scaled. Channel comparisons become easier. With Snap On you can select multiple time values in the [Scroll Bar](#). When the **Snap Arrow** is white  (Snap Mode turned off), the [values for channels in the main chart](#) are shown in absolute values where the max value of each channel graph is the same regardless of the position of the Viewport. Channel 33, which is snapped to the top of the chart in Snap Mode (shown above left), appears like the right image when Snap Mode is turned off.

- **Scrollbar Y-Axis Max** displays the maximum Y-Axis value in the [Scroll Bar](#).

4.4.5 Packet Error Rate - Sync Selected Packets With Other Windows



By default, and unlike other windows, PER Stats is not synchronized with other windows such as [Frame Display](#) in that selecting a frame range in one does not highlight the same frame range in the other. This ensures that **Frame Display** isn't constantly re-synchronizing during live capture while the view-port is maximized in PER Stats. If PER Stats synchronization is desired, it can be enabled by checking the **Sync Selected Packets with Other Windows** check box.

4.4.6 Packet Error Rate - Export

The Export section of PER Stats allows you to export data to a .csv or .txt file.

1. To use the Export, select a range of data using the [Viewport](#).
2. Select .csv or .txt from **Export Selected Data**, depending on what type of data file you want. The **Save As** dialog appears.

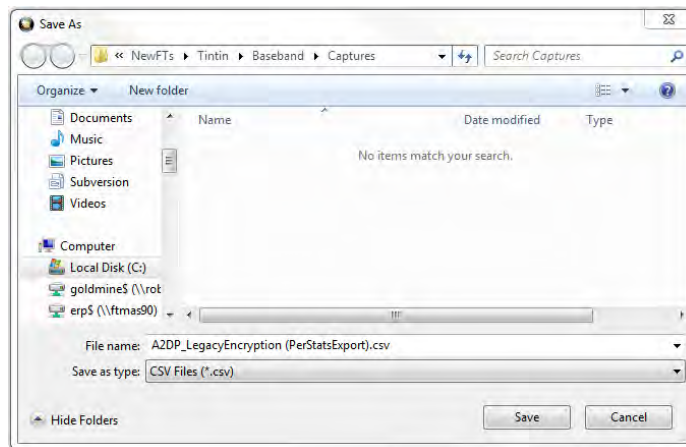
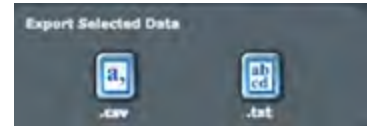


Figure 4.117 - Save As dialog in PER Stats Export

3. Select a location where you want to save the file in "Save in:".

4. Enter a file name in "File name:".
5. Select "Save".

The file will be saved to that location.

4.4.7 Packet Error Rate - Scroll Bar

The PER Stats **Scroll Bar** displays stats for all packets, divided into equal time intervals.

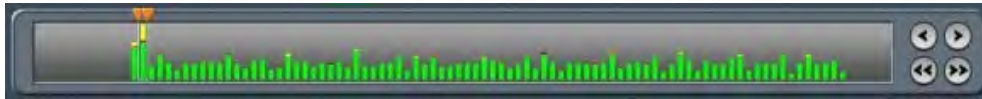

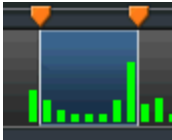






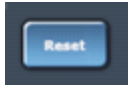


Figure 4.118 - PER Stats Scroll Bar

- Captured data begins to appear on the left and fills the width of the bar, left to right.
- The vertical bars in the **Scroll Bar** each indicate a fixed duration. When data first appears in the **Scroll Bar** as it is being captured, each bar equals one second. When the data fills the bar, reaching the right side limit, the last bar moves back to the center of the **Scroll Bar**. The bars stay the same size, but doubles in duration (for example, the first time the **Scroll Bar** fills, the bars return to the middle, but now each bar represent two seconds of time instead of one). Each time the bars cycle to the middle, the time they represent doubles. When the bars move and the **Viewport** (see below) is not maximized, the **Viewport** moves with the bars so that the same packet range is indicated. When the **Viewport** is maximized it stays maximized regardless of what the bars do. This ensures that the display can be made to reflect all packets at all times by maximizing the

- The **Viewport** is used to select single  or multiple vertical bars .
- You can drag the sides of the **Viewport** or the slider buttons to select multiple bars, representing a greater time range.
- You can click and drag the **Viewport** within the **Scroll Bar**.
- When you select a packet range in **Frame Display** that includes only some of the frames in PER Stats, the **Viewport** snaps up against the side of the bar with the unselected frames .
- When you select a packet range in Frame Display that includes all of the frames in PER Stats, the Viewport displays a space between the Viewport sides and the bar .
- Double clicking anywhere inside the **Scroll Bar** selects the entire **Scroll Bar**. Double clicking again toggles back to the previous size of the **Viewport**.
- Selecting Ctrl+A is the same as double-clicking.
- Clicking on a vertical bar left justifies the **Viewport** to that bar.
- Shift-clicking on a bar extends the nearest **Viewport** side to include that bar.

- The Home key moves the **Viewport** to the left edge.
- The End key moves the **Viewport** to the right edge.
- Pressing the left arrow button , the left arrow key, or the up arrow key moves the **Viewport** to the left, one vertical bar at a time.
- Pressing the right arrow button , the right arrow key, or the down arrow key moves the **Viewport** to the right, one vertical bar at a time.
- Pressing the double left arrow button  or the PgUp key moves the **Viewport** to the left by the current width of the **Viewport**. Holding down the Shift key will prevent the **Viewport** from moving if there is not enough room to move by its full width.
- Pressing the double right arrow button  or the PgDn key moves the **Viewport** to the right by the current width of the **Viewport**. Holding down the Shift key will prevent the **Viewport** from moving if there is not enough room to move by its full width.
- Holding the Shift key down and the right or left arrows moves the right side of the **Viewport**.
- Holding the Ctrl key down and the right or left arrows moves the left side of the **Viewport**.
- The Scroll bar includes inapplicable packets (sniffer debug, WiFi, etc) so that the packet range selected in [Frame Display](#) can be shown. Inapplicable packets are not, however, included in the [statistics reports](#).
- If the **Viewport** is adjusted within PER Stats, as opposed to selecting a packet range in [Frame Display](#), it uses only whole bars on both sides.
- Statistics are retained for all packets regardless of whether any of those packets have wrapped out. You can select the **Reset** button , which is located above the right portion of the **Scroll Bar**, to discard all stats for packets received up to that point.
- The **Reset** button is only available when you are capturing data.

4.4.8 Packet Error Rate - Excluded Packets

ID packets and packets that are missing channel numbers (such as HCI and BTSnoop) will not display data. ID packets are excluded because they can not have errors or indicate retransmission and therefore dilute the percentages for other packet types. Packets without channel numbers are excluded because the graphs are channel-specific. Before packets are captured, the Scroll Bar in Classic *Bluetooth* PER Stats contains the message "ID packets and packets without a channel number (such as HCI) are excluded", and the Scroll Bar in *Bluetooth* low energy PER Stats contains the message "Packets without a channel number (such as HCI) are excluded".

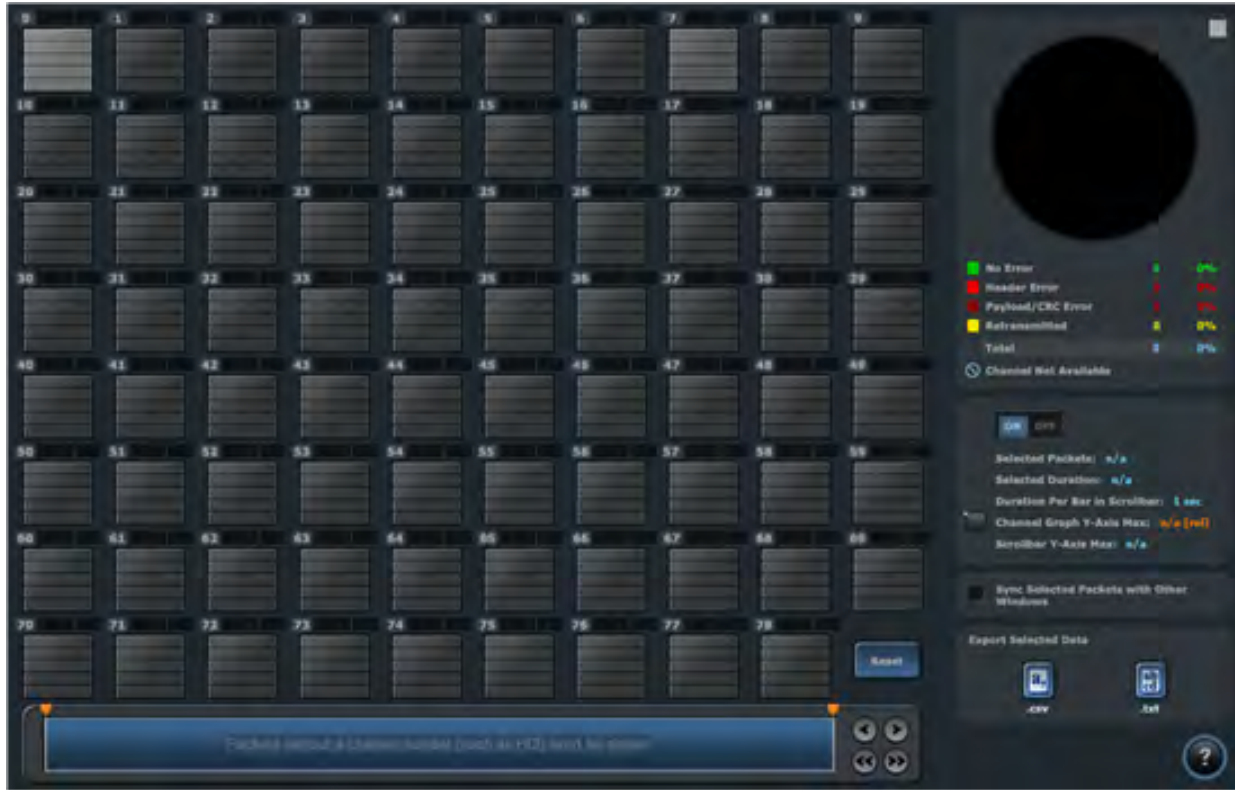


Figure 4.119 - Example: Excluded Packets Message in Scroll Bar (Classic Bluetooth)

4.5 Bluetooth Audio Expert System™



The *Bluetooth Audio Expert System™* monitors and analyzes *Bluetooth* audio streams with the purpose of detecting and reporting audio impairments. The primary goal of the Audio Expert System™ is to expedite the detection and resolution of *Bluetooth* protocol related audio impairments. To achieve this, the system automatically identifies audio impairments and reports them to a user as “events”. It also correlates the audio events with any detected codec

or *Bluetooth* protocol anomalies (events). The system allows a user to view the audio waveform, audio events, codec events, and *Bluetooth* protocol events on a time-aligned display.

An Audio Expert System™ event identifies to the user information, warnings, and errors. Event categories are shown in the following table.

Table 4.16 - Audio Expert System™ General Events

| Event Category | General Events Reported |
|--------------------|--------------------------|
| Bluetooth Protocol | Protocol violations |
| | Best practice violations |
| Codec | Configuration changes |
| | errors |

Table 4.16 - Audio Expert System™ General Events(continued)

| Event Category | General Events Reported |
|----------------|-------------------------|
| Audio | impairments (errors) |
| | information data |

When the Frontline software captures data, if there is audio content that must be debugged this data must be systematically examined when looking for the problem source. The effort to identify and correlate the audio related data can be daunting because the problem source may be caused by protocol, codec, or the audio itself. Using the Audio Expert System™ identifies events that are likely candidates for audio root cause analysis. The expert system examines all captured frames—in live capture or in capture file viewer—and selects audio-related protocol, codec, and audio events. The events are time correlated to the audio stream and identified with specific frames. In general, a cluster of events suggests an area for investigation, and in the presence of multiple event clusters the cluster with the most events suggests the best starting point.

The expert system works in conjunction with Frontline software that is operating in live capture mode or in capture file viewer mode. Selecting an event in the Audio Expert System™ will simultaneously highlight related packets in the Frontline software **Frame Display, Coexistence View, Message Sequence Chart, Bluetooth Timeline, and Packet Error Rate Statistics (PER Stats)** windows.

Audio Expert System™ further provides methods for isolating testing to specific audio events by using two operating modes: non-referenced and referenced.

Table 4.17 - Audio Expert System Operating Modes

| Mode | Description |
|----------------|--|
| Non-referenced | Processing audio of completely unknown program content (e.g. arbitrary music or speech content). Since the system does not have any prior knowledge of the audio being analyzed, the types of audio analysis that can be performed is limited. |
| Referenced | A “pseudo closed loop” test scenario where the user plays specific Reference Audio files (pre-recorded audio test files provided by Frontline) on the Source DUT (Device Under test). The analysis of the received audio results in a series of “Audio Events” being reported by comparing changes in the received audio to expected changes of the Reference Audio, and reporting deviation events when they occur. |

Reference mode detects a larger number of events because the reference audio has specific frequency, amplitude, and duration occurring at known points in time allowing for precise comparison.

4.5.1 Supported Codec Parameters

Supported Parameters for SBC Codec

- Sampling Frequencies: 16 KHz*, 32 KHz*, 44.1 KHz, 48 KHz
- Channel Modes: Mono, Dual Channel, Stereo, Joint Stereo
- Block Length: 4, 8, 12, 16

- Number of subbands: 4, 8
- Allocation Method: SNR, Loudness
- Minimum Bitpool Value: 2
- Maximum Bitpool Value: 53

Supported Parameters for MPEG-2, 4 AAC

- Object Types; MPEG-4 AAC LC
- Sampling Frequencies: 44.1 KHz, 48 KHz, 8 KHz*, 11.025 KHz*, 12 KHz*, 16 KHz*, 22.050 KHz*, 24 KHz*, 32 KHz*, 64 KHz*, 88.2 KHz*, 96 KHz*
- Channels: 1 and 2
- Variable Bit Rate and Specified Bit rate

* Audio Analysis not supported . Although, user will be able to play back the audio live.

Supported Parameters for aptX

- Object Types; aptX-classic, aptX-LL (both content protected and non-content protected)
- Audio Format: 16-bit, 44.1kHz
- Data Rates: 352 kbps

Supported Parameters for CVSD

- Channel Mode: Mono
- Sampling Rate: 64 kHz

Supported Parameters for mSBC codec

- Channel Mode: Mono
- Sampling Rate: 16 kHz
- Allocation method: Loudness
- Subbands: 8
- Block Length: 15
- Bitpool: 26

4.5.2 Starting the AudioExpert System (BPA 600 only)


To use the Audio Expert System™, the user must have

- Current Premium Maintenance purchased from Frontline
- Frontline hardware, with Audio Expert System™ license installed, connected to the PC. This is a requirement for both live capture and when viewing a saved capture file.

For live capture, set up the Frontline BPA 600 datasource and begin capturing data.

Note: Proper positioning of the Frontline hardware relative to the devices under test (DUT1-source, DUT2-sink) will contribute to effective data capture. [Air Sniffing: Positioning Devices on page 73.](#)

For viewing a capture file, load the saved file from the **Control** window **File** menu.

When an audio stream is available the open the **Audio Expert System™ Window** by clicking on the **Control** window Audio Expert System™ button . If the Frontline hardware is not licensed for Audio Expert System™, the button will not be present.

4.5.3 Operating Modes

The *Bluetooth* audio analysis can be accomplished in two modes: 1) unreferenced mode, and 2) referenced mode.

4.5.3.1 Non-Referenced Mode

In Non-Referenced Mode, the system is typically processing audio of completely unknown program content (e.g. arbitrary music or speech content). Since the system does not have any prior knowledge of the audio being analyzed, the types of audio analysis that can be performed is limited.

The following events are reported whenever the system is operating in Non-Reference mode. These are the meaningful audio analysis that the system can perform without reporting too many false positive results.

- Volume Level (Low Volume or High Volume): Reported if the average volume level is not in a range conducive to performing meaningful audio analysis.
- Clipping: Amplitude distortion due to a signal amplitude exceeding the maximum value that can be represented by the digital system
- Dropout: Abrupt and very short duration intervals of silence
- Glitch: Extremely large sample-to-sample audio amplitude transitions that have little probability of occurring within natural speech or music

4.5.3.2 Referenced Mode

In Referenced Mode, the system operates in a “pseudo closed loop” test scenario where the user plays a specific Reference Audio file on the Source DUT. The Source DUT negotiates with the Sink DUT to determine the appropriate codec and audio parameters to use and will then process the Reference Audio file accordingly before transmitting the resulting audio via *Bluetooth*. The Reference Audio is a pre-recorded audio test file provided in the Frontline software installer.

The Sink DUT receives the encoded audio, decodes it, and processes it for playback. In parallel, the Frontline analyzer unit snoops the over-the-air signal between the Source DUT and Sink DUT and emulates the RF reception and decoding done inside the Sink DUT. The Audio Expert System™ automatically detects that a Reference Audio file is being received and then analyzes the resulting audio for deviations from expected parameters.

Referenced Audio files are protocol specific.

The following events are reported whenever the system is operating in the Referenced mode.

- Test ID Found
- Test Script Not Found
- Invalid Test Script
- Synchronization Lost
- Unexpected Frequency
- Unexpected Level
- Unexpected Duration
- Amplitude Fluctuation
- Unexpected Phase Change
- Clipping
- Excess Noise
- CVSD HF Level Too High
- End of Test

Reference Audio Test Files

The Reference Audio files are specific audio files that exercise the system so that audio impairments can more efficiently and accurately be identified and reported. The Reference Audio files are composed of a series of back-to-back and relatively short duration tones of changing amplitude, frequency, and duration.

The test files are stored on the users computer In the directory "`\Frontline <version #>\Development Tools\Audio Expert Test Files\`". For example,

`Test_1.03_48kHz_16Bit_3Loops_2Ch.wav`

Note: Reference test files are periodically updated. Shown here is an example. Files delivered with your latest Frontline software version may have changed. Contact Frontline Technical Support for information on the latest reference file versions.

The test files have a set of tones forming a unique Test ID that lets the ComProbe analyzer know that it is capturing a test file instead of an arbitrary audio stream. There is no need for special configuration of the ComProbe analyzer. The Test ID will have the identifier notation N.vv, where N = the file number and vv = a two digit version, for example 1.02.

Using the Test Files

The analysis of the received audio results in a series of Audio Events being reported by comparing changes in the received audio to expected changes of the Reference Audio, and reporting deviation events when they occur.

The system starts up in Non-Referenced mode, and is continuously looking for a valid Reference Audio file by measuring frequency and amplitude of the received over-the-air audio. Transitioning to Referenced mode requires the successful detection of a Test ID tone sequence of proper frequency, duration, and value.

Once the Referenced Mode state is achieved, the expectation is that all tones encountered will conform to the script identified by the Collected Digits (the "Test ID"). The system remains in the Referenced Mode state until either the end of test is reached, or a loss of synchronization occurs.

The synchronization of the received audio (from the Reference Audio files) versus the internal Test Script is achieved based on changes in frequency of the tones in the Reference Audio file. Frequency changes are used because this parameter is relatively immune to the configuration of the network.

For a comparison of reference mode detectable problems to unreferenced detectable problems see the table in [the audio event type table](#).

The Test Script

The Reference Audio used for Referenced Mode testing is generated from scripts that define a series of audio segments. Each segment provides an audio tone parameters including frequency, amplitude, duration, fade in and fade out durations, and start time. The script is an XML file delivered with the Frontline software. This file is used during Referenced mode testing for comparison to the "sniffed" Reference Audio parameters of frequency, amplitude, duration, etc.

Below is a sample script table and the resulting sample Reference Audio .wav file. The generated .wav file begins with a Test ID that is used to identify the "sniffed" audio as a Reference Audio file, and the Audio Expert System™ automatically switches from Non-Referenced mode to Referenced mode.

```
<?xml version="1.0" encoding="UTF-8"?>
- <SegmentArray>
  - <Segment>
    <SegID>0</SegID>
    <Opcode>F</Opcode>
    <Frequency>100</Frequency>
    <Level>-95</Level>
    <Cycles>10</Cycles>
    <Duration>0.1</Duration>
    <FadeIn>0</FadeIn>
    <FadeOut>0</FadeOut>
    <StartTime>0</StartTime>
  </Segment>
  - <Segment>
    <SegID>1</SegID>
    <Opcode>F</Opcode>
    <Frequency>210</Frequency>
    <Level>-3</Level>
    <Cycles>21</Cycles>
    <Duration>0.1</Duration>
    <FadeIn>0</FadeIn>
    <FadeOut>0</FadeOut>
    <StartTime>0.1</StartTime>
  </Segment>
  - <Segment>
    <SegID>2</SegID>
    <Opcode>F</Opcode>
```

Table 4.18 - Sample Test Script Table

| Segment | OpCode | Frequency | Level | Cycles | Duration | Fade in | Fade Out | Start Time |
|---------|--------|-----------|-------|--------|----------|---------|----------|------------|
| 1 | F | 200 | 0 | 5 | 0.025 | 0 | 0 | 0.000 |
| 2 | F | 1000 | 0 | 25 | 0.025 | 0 | 0 | 0.025 |
| 3 | F | 300 | -12 | 15 | 0.050 | 0 | 0 | 0.050 |
| 4 | F | 600 | 0 | 30 | 0.050 | 0 | 0 | 0.100 |
| 5 | F+ | 880 | -6 | 44 | 0.050 | 0 | 0 | 0.150 |
| 6 | F+ | 240 | -6 | 12 | 0.050 | 0 | 0 | 0.150 |
| 7 | F | 600 | -95 | 30 | 0.050 | 0 | 0 | 0.200 |
| 8 | F | 600 | 0 | 30 | 0.050 | 0 | 10 | 0.200 |

4.5.3.3 Referenced Mode Testing Processes

In the Referenced mode, the devices under test use a specific audio file (called reference file or test file) provided by Frontline whose contents are already known to the Frontline software. The software compares the parameters of the received audio data against its parameters and presents analysis for the user. Commonly, in Bluetooth technology the music sent via A2DP and speech sent via HFP. There are a few ways users can conduct referenced mode testing depending upon what profile they are using. The figure 17 shows the source of the audio and the medium through which it can be accessed by Source device to send to sink device via Bluetooth.

Table 4.19 - Referenced Mode Testing Process Between Two DUTs

| Audio Source | Process to Send Using A2DP | Process to Send Using HFP |
|--|--|---|
| A file stored on the device's local memory | Play the locally stored file on the audio source device | Play using the third party App that transmits music data on HFP. |
| Streaming audio over a cellular network | Play the test in a browser on the audio source device https://youtu.be/rmirDbikrtM | Make a call to 434-964-1407 or 434-964-1304 through a cellular network. The phone number receiving the call playbacks recorded test signal. |
| Streaming audio over a Wi-Fi network | Play the test in a browser on the audio source device https://youtu.be/rmirDbikrtM | Make a call to 434-964-1407 or 434-964-1304 through a VoIP provider such as Skype. The phone number receiving the call playbacks recorded test signal. Potential problem: The VoIP provider might use custom codecs and cause undesirable behavior. |

A2DP

Playing the test file locally

The simplest way to perform music data testing is to directly play the reference file from DUT1 to DUT2. To do that, save the reference file provided with the Frontline software on the Source device. Then connect the Bluetooth enabled devices and play the music file from one device to the other. The software will automatically detect the mode and present analysis for the user.

Playing the test file via Internet

If the user is testing a scenario where they need to analyze audio played through the internet (either using Wi-Fi or cellular data plan), they may access the reference file on YouTube provided by Frontline - <https://youtu.be/rmirDbikrtM>. Note that the software is only analyzing the Bluetooth link between the two DUTs. Any abnormalities at the Wi-Fi and cellular network level will affect the audio quality that may not be Bluetooth protocol related and the software will not be able to detect that.

HFP

Playing the test file by calling a phone number

Frontline provides the following phone numbers - 434-964-1407 and 434-964-1304 that users can call, to conduct speech audio data analysis over Bluetooth. The calls can be made using the cellular network (most common method) or VoIP. Again, the VoIP provider might use custom codecs and cause undesirable behavior which cannot be detected by Audio Expert System™ software.

Playing the test file using Third party Apps

Bluetooth Audio Expert System™ Reference mode testing can be accomplished using third party apps on Android, iOS, and Windows phones. The following apps are available from their respective App stores:

- [BTmono, Android](#)
- [Blue2Car, IOS](#)
- [Windows Headset player lite](#)


Note: When selecting and using these apps, thoroughly review all the vendor documentation. While Teledyne LeCroy has conducted testing of these apps, Teledyne LeCroy has not completed full interoperability testing with our library of *Bluetooth* devices and does not warrant the use of these apps with every device when using the following procedures. Teledyne LeCroy does not provide support or maintenance for third party apps. Any issues or questions should be directed to the app developer.

1. In the following steps Device Under Test 1 (DUT1) is the device sending the reference test file to DUT2.
2. Download the third party app to DUT1 and follow the app vendor's instructions for installation and use.
3. Load the Audio Expert System reference test file

"Test_1.02_64.1kHz_16Bit.wav"

on DUT1. The test file is stored on the users computer In the directory "\\Frontline <version #>\Development Tools\Audio Expert Test Files\".

Note: Reference test files are periodically updated. Shown here is an example. Files delivered with your latest Frontline software version may have changed. Contact Teledyne LeCroy Technical Support for information on the latest reference file versions.

4. With the BPA 600 connected to the computer, configure the datasource, and follow procedures to capture data.
5. Launch Audio Expert System by clicking on the **Control** window .
6. Turn on Bluetooth on your DUTs, DUT1 and DUT2. Turn on the third party *Bluetooth* app for routing the reference file over A2DP or HFP by following the vendor's directions.
7. Send the reference test file from DUT1 to DUT2 via the third party app.
8. Observe the events in the Audio Expert System™ **Events Table**. Look for an event **Description**:

“TestIDFound : REF: Test ID 1.02, Channel Gain = -11.8 dB TermFreq=400.0”.

Note: This is an example. The display may vary with the reference file version.

The Frontline analyzer has successfully detected the reference test signal and the system is locked into reference mode.

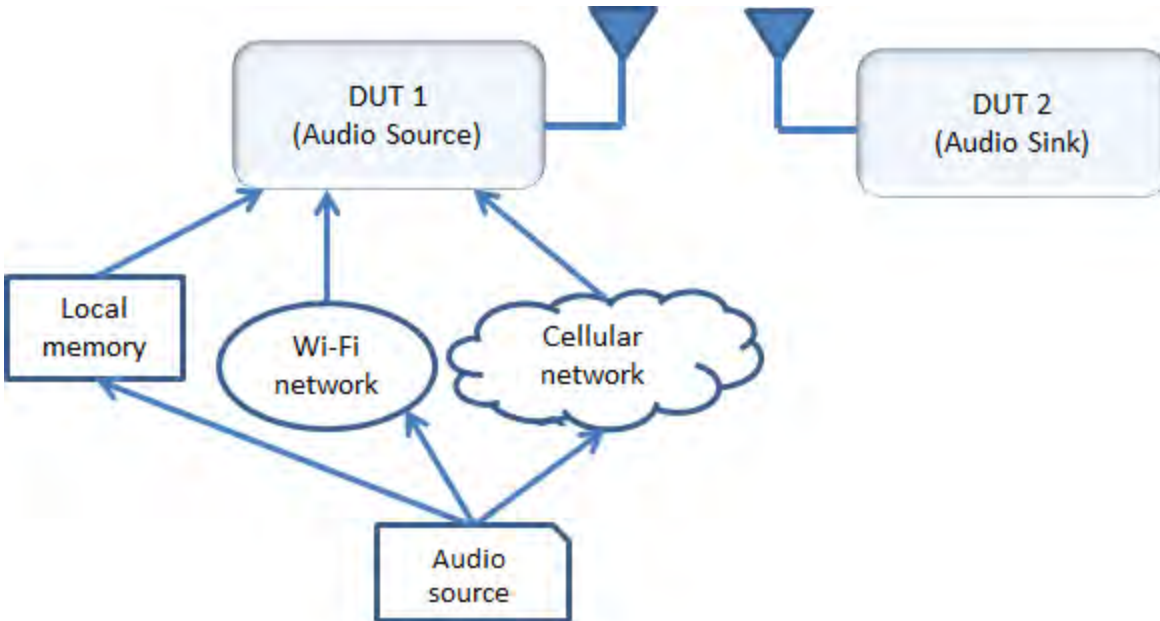


Figure 4.120 - Test Cases for Referenced Mode Testing

4.5.3.3.1 System Calibration for Referenced Mode

The objective is to achieve settings at the *Bluetooth* source device (DUT1) that bring the PCM sample levels of tones in the Reference Audio files sent over-the-air as close as possible to the levels at which they were created, without exceeding them. Test ID tones, and the tones in test file sequences for Referenced Mode are generally recorded with a maximum tone segment level of -3 dBFS, although there are a few exceptions where signal levels may be as high as -1 dBFS.

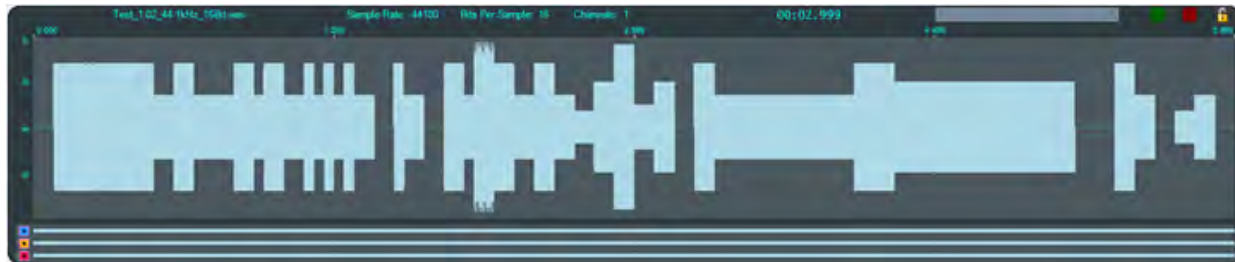


Figure 4.121 - Test_1.02_44.1kHz_16Bit.wav Waveform

Show in the image above, is a graphic of the overall envelope of the Reference Audio test file “Test_1.02_44.1kHz_16Bit.wav. Test 1.02 is a test file that enables a wide range of tests that includes a number amplitude changes, frequency changes, intentional silence, and multi-frequency tone segments. Its goal is to flush out the audio chain’s general ability to convey amplitude, frequency, silence, and duration.

The ideal calibration for this file is one where the waveform visualization on Frontline’s Expert System User Interface (UI) looks identical to the one shown below with respect to maximum levels. In particular, there are three segments in this test whose peaks are at exactly -6 dBFS. That is, there is zero loss or gain through the chain.

Table 4.20 - Test 1.02 -6 dBFS Segments

| SegmentID | Frequency, Hz | Start Time, sec. | Duration, sec. |
|-----------|---------------|------------------|----------------|
| 32 | 800 | 2.800 | 0.100 |
| 35 | 1120 | 3.100 | 0.100 |
| 40 | 400 | 4.300 | 0.900 |

These -6 dBFS segments are described in the Test 1.02 -6dBFS Segments table . These segments serve as a convenient and quick visual indicator that levels are appropriate, especially the longer 3rd case which is evident at the 4.999 second reference time of the above image(a little over 2/3 of the way through the test).

The first 0.500 seconds of Test 1.02, which contains the Test ID value “1.02” is shown below. The three digits ‘1’, ‘0’, and ‘2’ are represented by the low frequencies 210Hz, 200Hz, and 220Hz, respectively, which are 100 milliseconds in duration, and are separated by 1 kHz digit delimiters of 50 milliseconds duration. The final tone is a 100 millisecond segment at 400 Hz, defined as a “Test ID Terminator”. Note that since the levels of all of these tones are at exactly -3 dBFS, the peak levels should be exactly halfway between any available -6 dBFS (50%) gridline.

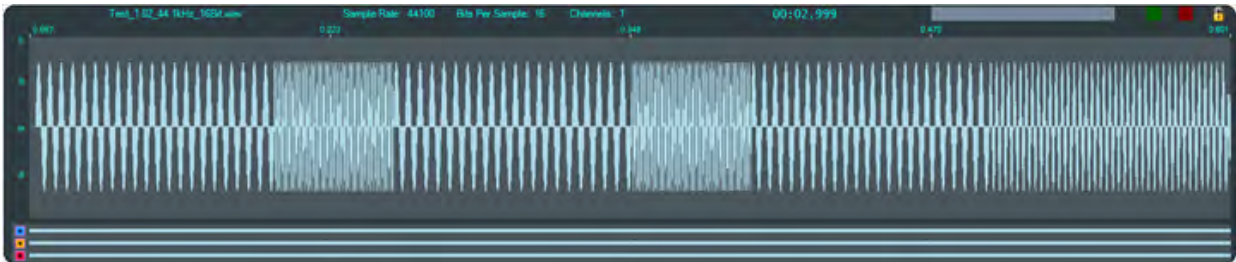


Figure 4.122 - Test 1.02 Test ID Segment

The three digits ‘1’, ‘0’, and ‘2’ are represented by the low frequencies 210 Hz, 200 Hz, and 220 Hz, respectively, which are 100 ms in duration, and are separated by 1 kHz digit delimiters of 50 ms duration. The final tone is a 100 ms segment at 400 Hz, defined as a “Test ID Terminator”. Note that since the levels of all of these tones are at exactly -3 dBFS, the peak levels -3 dBFS.

The value in the Info1 parameter of the “Test ID Found” event is optimally the value 23196 and may be converted to dBFS by the relationship

$$\text{dBFS} = 20 \log_{10} \left(\frac{\text{info1}}{32767.0} \right)$$

Optionally the value can be interpreted as “Channel Gain” via the relationship

$$\text{dB} = 20 \log_{10} \left(\frac{\text{info1}}{23196.0} \right)$$

Table 4.21 - "Test ID Found" Event "info1"
Maximum and Minimum Values

| Format | Application | Maximum | Minimum |
|-------------|-------------|---------|----------|
| Integer | Speech | 23196 | 5826 |
| | Music | 23196 | 3297 |
| Level | Speech | -3 dBFS | -15 dBFS |
| | Music | -3 dBFS | -20 dBFS |
| Chanel Gain | Speech | 0 dB | -12 dB |
| | Music | 0 dB | -17 dB |

This table indicates the maximum and minimum acceptable levels for the "Test ID Found" Info1 parameter in integer form, decibel level in dBFS, and Channel Gain in dB.

Example 1: For the case where the Info1 parameter is converted to "Channel Gain", if the audio is speech (i.e. transported via a SCO channel), then a value of -11.9 dB is acceptable, and a value of -12.1 dB is not.

Example 2: For the case where the Info1 parameter is converted to "Channel Gain", if the audio is music (i.e. transported via an A2DP connection), then a value of -16.9 dB is acceptable, and a value of -17.1 dB is not.

For both cases, at the high volume end, a value of -0.1 dB is acceptable, a value of 0.1 dB is not.

The dynamic range of the audio path is important to understand because it has a direct impact on measurement accuracy. Only levels at or above the minimum and at or below the maximum are examined for expected level and frequency.

4.5.3.3.2 Adjusting for Optimal Volume Levels

The exact steps that need to be taken depend on the exact devices being used, and their device specific setup requirements, and the speech or audio configuration under test. For the simplest case where, for example, a "music" audio file is to be played by a smartphone to a set of *Bluetooth* speakers, the typical steps would include the following.

1. Choose an audio reference file to be played at DUT1 appropriate for the configuration to be tested.

The test files are stored on the users computer In the directory "\\Frontline <version #>\Development Tools\Audio Expert Test Files\". For example,

Test_1.03_48kHz_16Bit_3Loops_2Ch.wav

Note: Reference test files are periodically updated. Shown here is an example. Files delivered with your latest Frontline software version may have changed. Contact Frontline Technical Support for information on the latest reference file versions.

2. Before establishing the *Bluetooth* connection, play the file while listening to it on the DUT1 device itself, and become familiar with the overall sound quality, generally ignoring exact volume.
3. Set the playback volume at DUT1 to maximum.
4. Set the playback volume at DUT2 to minimum.
5. Establish the *Bluetooth* connection and begin playback of the file on DUT1, if possible in "Loop" or "Repeat" mode to avoid having to continuously restart.
6. Slowly increase the volume on DUT2 until it is at a comfortable level.
7. If the audio sounds distorted, reduce the playback volume at DUT1, and repeat Step 6.

8. When the clarity of the audio is comparable to that heard when listening to the DUT1 device, proceed with using the Frontline software enabled to capture and analyze the Bluetooth data.
9. Visually observe the waveform in the Audio Expert System **Wave Panel** comparing it to the image above, Figure 1.1. If the level of the -6 dB, 0.9 sec duration, 400 Hz tone (a little over 2/3 of the way through the test) is grossly above or below the -6 dB (50% volume) grid line, adjust the DUT1 volume accordingly and repeat this step. Optimally it would be on or just below the -6 dB gridline, but not above. The peak should never hit the maximum positive or negative limits of the display.
10. Find the “Test ID Found” event in the **Event Table** to verify that the system has transitioned to Referenced Mode, and verify that the value for “Channel Gain” (or “Level” as implemented in the UI) is within the range of values specified in Table 1-2.

If the observed (captured) waveforms do not reasonably conform to the above graphic for Test_1.02, or the “Test ID Found” event is not reported, there is a problem along the audio chain. This could be as simple as a configuration setting, or more subtle such as an encoder/decoder incompatibility.

4.5.4 Audio Expert System™ Event Type

The following tables list the Audio Expert System™ *Bluetooth*, *Codec*, and audio events with description. Included in the tables is the event severity that can have three values: Information, Warning, and Error. The event severity will appear as icons and text in the Audio Event System once an audio streams has been captured. Refer to [4.5.5.3 Event Table](#), [Event Table Columns on page 245](#) for an explanation of the severity types.

4.5.4.1 Event Type: *Bluetooth Protocol*

Table 4.22 - Event Type: *Bluetooth Protocol*

| Protocol | Severity | Description |
|----------|----------|---|
| A2DP | Warning | AVDTP signal response received for unknown command. |
| A2DP | Warning | Unrecognized capability type |
| A2DP | Error | eSCO parameters requested. |
| A2DP | Error | Profile TX PDUs larger than available bandwidth for active A2DP Streaming interval. |
| A2DP | Error | Bitpool value does not match configured bitpool range. |
| A2DP | Error | Attempt to suspend inactive stream. |
| A2DP | Error | Configuration attempt using unsupported CODEC. |
| A2DP | Error | Incorrect AVTDP command length. |
| A2DP | Error | Unknown command Stream End Point Identifier (SEID). |
| A2DP | Error | A2DP stream configuration attempt using invalid CODEC parameters. |
| A2DP | Error | A2DP stream configuration request sent during active stream. |
| A2DP | Error | Audio data length does not match length header. |
| A2DP | Error | Incorrect A2DP SBC frame fragmentation. |
| A2DP | Error | A2DP SBC frame header contents does not match stream configuration. |
| A2DP | Error | Attempt to configure A2DP stream with unsupported configuration. |

Table 4.22 - Event Type: Bluetooth Protocol(continued)

| Protocol | Severity | Description |
|----------|----------|--|
| A2DP | Error | Reported A2DP stream capabilities do not contain mandatory features. |
| A2DP | Error | A2DP streaming L2CAP channel not disconnected after ABORT operation. |
| A2DP | Error | Fragmented AVDTP packet not terminated before sending next packet. |
| A2DP | Error | Invalid AVDTP transaction ID. |
| A2DP | Error | Missing AVDTP command response. |
| A2DP | Error | Unrecognized A2DP content protection type. |
| A2DP | Error | Attempt to configure delay reporting during incorrect stream state. |
| A2DP | Error | Attempt to open A2DP stream that has not been configured. |
| A2DP | Error | Attempt to close A2DP stream that is not active. |
| A2DP | Error | A2DP streaming channel created before configuration completed. |
| A2DP | Error | Configuration command contains invalid length parameter. |
| A2DP | Error | Configuration command contains invalid media transport format. |
| A2DP | Error | SBC CRC Error. |
| A2DP | Error | SBC invalid channel mode. |
| A2DP | Error | SBC invalid header. |
| A2DP | Error | Invalid AVDTP configuration parameter. |
| A2DP | Error | Invalid AVDTP stream state |

4.5.4.2 Event Type: Codec

Table 4.23 - Event Type: Codec

| Codec | Severity | Event | Description |
|-------|-------------|----------------------------------|--|
| SBC | Information | Codec Initialization | Codec session started |
| SBC | Information | Codec tear-down | Codec session ended |
| SBC | Information | Stream Re-configuration | Stream Re-configuration |
| SBC | Error | Incorrect Configuration Detected | SBC Codec detected a change in audio parameters |
| SBC | Error | Lost Sync | SBC Codec expected to find synch word: 0x9C instead found: 0x: typically due to corrupted data |
| SBC | Error | Bad Header | SBC Codec detected corrupted header: typically due to corrupted data |
| SBC | Error | CRC Failure | SBC Codec detected bad CRC: typically due to corrupted data |

Table 4.23 - Event Type: Codec(continued)

| Codec | Severity | Event | Description |
|-------|-------------|--|--|
| SBC | Error | No output | SBC Codec generated no output due to corrupted data |
| mSBC | Information | Codec tear-down | Codec Session Ended |
| mSBC | Information | Stream Re-configuration | Stream Re-configuration |
| mSBC | Warning | Packet Loss Concealment | mSBC Codec detected a bad frame and generated substitute data to compensate for it |
| mSBC | Error | Incorrect Configuration Detected | mSBC Codec detected a change in audio parameters |
| mSBC | Error | Lost Sync | mSBC Codec expected to find synch word: 0xAD instead found: 0x: typically due to corrupted data |
| mSBC | Error | Bad Header | mSBC Codec detected corrupted header: typically due to corrupted data |
| mSBC | Error | CRC Failure | mSBC Codec detected bad CRC: typically due to corrupted data |
| mSBC | Error | No output | mSBC Codec generated no output due to corrupted data when PLC not configured |
| AAC | Information | Codec initialization | Codec session started |
| AAC | Information | Codec tear-down | Codec session ended |
| AAC | Information | Bitstream type set | The bitstream type has been set. For Bluetooth, it should be LATM. |
| AAC | Warning | Single frame error, concealment triggered. | During decoding, a single frame error was detected which triggered built in concealment processing. |
| AAC | Error | Codec setting change | The codec has been re-initialized due to a setting change. |
| AAC | Error | Unframed stream error | A frame error was detected for an unframed stream. The codec is being reset in order to continue processing. |
| AAC | Error | Transport not initialized | The codec cannot be initialized for the given transport. |
| AAC | Error | Transport not supported | The selected transport is not supported. This could occur when an out of band LATM is selected opposed to in band. |
| AAC | Error | Transport failure | General failure in the transport. |
| AAC | Error | Transport error | This typically occurs when there isn't any configuration information available. |

Table 4.23 - Event Type: Codec(continued)

| Codec | Severity | Event | Description |
|-------|-------------|----------------------|---|
| AptX | Information | Codec initialization | Codec session started |
| AptX | Information | Codec tear-down | Codec session ended |
| AptX | Error | Bad Data | Non-stereo data has been detected for incoming data stream. |

4.5.4.3 Event Type: Audio

Table 4.24 - Event Type: Audio

| Test Mode | Severity | Event | Description |
|----------------|----------|--------------------------|---|
| Non-Referenced | Warning | Low Volume Alarm | Warn the user that the volume level of the detected audio is below the best range for performing meaningful audio analysis. Alarm is initialized when volume level above the “ Measurement Threshold ¹ ” level is detected. Alarm is activated when the detected volume drops below the “Measurement Threshold” level for 10 consecutive 0.5 sec measurement intervals. |
| Non-Referenced | Warning | Clipping | Reports the detection of suspected distortion that occurs when the amplitude of a signal exceeds a digital systems ability to represent it accurately. Clipping is a type of amplitude distortion. The system reports a Clipping event when consecutive samples at the maximum value that can be represented by the digital system have been detected. Note that the maximum value that can be represented is different depending on the number of bits per sample (i.e. bits of resolution) of the audio stream. The system limits the number of reported Clipping events to typically 10 to 20 per sec. |
| Non-Referenced | Warning | High Volume Alarm | Warn the user that the volume level of the detected audio is above the best range for performing meaningful audio analysis (i.e. above a level where the audio will likely become distorted). Alarm is activated when the detected audio volume is continuously above the high volume threshold ² (see Figure 2) for 10 consecutive 0.5 sec measurement intervals (i.e. 5 sec total). The event will not be repeated again until the detected volume level drops below the high volume threshold for 10 more consecutive 0.5 sec measurement connections. |

¹The volume threshold above which useful audio analysis is possible.

²High Volume Threshold for speech: - 6dBFS High Volume Threshold for music: -12 dBFS

Table 4.24 - Event Type: Audio(continued)

| Test Mode | Severity | Event | Description |
|----------------|----------|-------------------------|--|
| Non-Referenced | Warning | Dropout | Reports the detection of an unusual brief silence period where the brief silence is preceded and followed by “normal” audio levels. A typical definition of Dropout is the short dramatic loss of volume typically caused by lost digital information. Root causes include transmission system errors resulting in lost data packets, transmission channel reconfigurations, bad sections of memory, processor overloads that temporarily interrupt the flow of information, and so on. |
| Non-Referenced | Warning | Glitch | Extremely large sample-to-sample audio amplitude transitions ¹ that have little probability of occurring within natural speech or music. Such dramatic changes would typically happen only in situations of dropped samples. |
| Referenced | Info | TestID Found | Occurs when a valid Test ID ² has been recognized. A valid Test ID must meet the level, frequency, duration, and delimiter requirements. If any of these parameters do not match, the process is terminated and is reset to the initial conditions. Until a Test ID is successfully recognized, the system will continue to operate in Non Referenced Mode; therefore, no events related to false starts are reported. This is because for arbitrary audio there is no expectation of any Test ID. |
| Referenced | Warning | Test Script Not Found | Occurs if a valid Test ID was found , but the script for that Test ID was not found. The system reverts to Non-Referenced Mode if this happens. This event should not occur if using a valid Reference Audio file provided by Frontline. |
| Referenced | Error | Invalid Test Script | This event is generated when an error occurs while accessing information in a script. This event should not occur if using a valid reference audio file provided by Frontline. |

¹Glitch sample-to-sample audio amplitude transits: Speech: greater than 40 dB change Music: greater than 90 dB change

²A “Test ID” is three digits minimum in length, representing a dot notation “N.w” Test Identifier. The Value ‘N’ may be any length ≥ 1 indicating a specific test number, and “w” represents a two digit version. Each digit is represented by a tone between 200 and 290 Hz, and is followed either by a 1 kHz delimiter tone or a 400 Hz Test ID terminator. The digit ‘0’ is represented by 200 Hz, the digit ‘1’ by 210 Hz, and so on, up to the digit ‘9’ represented by 290 Hz.

Table 4.24 - Event Type: Audio(continued)

| Test Mode | Severity | Event | Description |
|------------|----------|----------------------|---|
| Referenced | Error | Synchronization Lost | Generated when after a successful TestID recognition the system encounters unexpected frequencies or durations of audio segments while analyzing a received Reference Audio file. If this situation occurs, the internal segment tracking logic attempts to look forward and/or backward in the test script to determine if the currently measured characteristics are consistent with the previous or next segment of the script. If there is a match, the internal segment pointer is advanced or retarded appropriately, the Synchronization Lost event is not generated, and the audio analysis continues. However, if a match cannot be found, the system declares itself out of sync and generates the Synchronization Lost Event, terminates any active test script, and reverts to Non-Referenced Mode. |
| Referenced | Error | Unexpected Frequency | Reported when a measured frequency deviates from an expected frequency by a specific percentage (determined by the negotiated parameters of the over-the-air audio stream). The system knows the Reference Audio file that is being played on the Source DUT; therefore, the system knows which frequencies (tones) to expect at a given time. |
| Referenced | Error | Unexpected Level | Reported when the measured level at the start of a tone segment is not within tolerance. The tolerance is dependent on sample rate and bits per sample, but it generally is +/- 3 dB for speech and +/-11 dB for music. The system knows the Reference Audio file that is being played on the Source DUT; therefore, the system knows which amplitude level to expect at a given time. |

Table 4.24 - Event Type: Audio(continued)

| Test Mode | Severity | Event | Description |
|------------|----------|-------------------------|--|
| Referenced | Error | Unexpected Duration | Reported when a tone segment of the Reference Audio file is shorter or longer than expected ¹ . The system knows the Reference Audio file that is being played on the Source DUT and therefore knows how long a specific tone segment should last. If either a change of amplitude or frequency arrives either before or after that programmed duration, then the change is by definition unexpected. This type of audio impairment can be caused by lost or corrupted data, repeated data, faulty packet loss concealment algorithms, etc. |
| Referenced | Error | Amplitude Fluctuations | Reported if the system detects unexpected amplitude changes over a given interval. The test tones in Frontline's Reference Audio files have a fixed amplitude level over their duration. Therefore, if the corresponding audio levels received over the air by the system fluctuates ² more than a specified level (this level is based on the received audio stream parameters), then the system generates an Amplitude Fluctuations event. |
| Referenced | Error | Unexpected Phase Change | Provides a fine-grained indication of lost or repeated energy. The system knows when a specific tone should be expected. During this interval, the system checks that the measured average frequency is the same as the expected frequency. If this is correct, the system will continue to monitor the instantaneous frequency. If the instantaneous frequency deviates sufficiently from the current average frequency, the frequency measurement state machine will reset and begin re-measuring. Typically, the outcome is the discovery of the next scripted (expected) frequency. However, another outcome can be that the same frequency as the previous average frequency is rediscovered, and this is reported as an Unexpected Phase Change event. Such phase changes are an indicator of losses of signal that do not result in amplitude dropouts, or signal substitution (repetition) of previous audio energy due to things such as "packet loss concealment" tactics. |

¹The amount that a measured duration must deviate from the programmed duration of a tone segment before the system declares this event varies, depending on the negotiated over-the-air audio stream specific parameters, but it is generally in the range of 5% to 10%. Note that this event will result in an attempt to resynchronize if the measured duration is greater than expected.

²The system calculates amplitude fluctuations as: $(\text{Max Level} - \text{Min Level}) / (\text{Max Level} + \text{Min Level}) * 100$

Table 4.24 - Event Type: Audio(continued)

| Test Mode | Severity | Event | Description |
|------------|----------|--------------------------|---|
| Referenced | Error | Excess Noise | The Excess Noise event is reported when energy sufficiently above the "Silence Threshold" is detected during programmed segments of silence. Excess noise can indicate a poor analog audio chain with an inherently poor noise floor, glitches occurring during silence intervals, or codecs that do not transition to silence instantaneously. |
| Referenced | Error | Clipping | Reports the detection of suspected distortion that occurs when the amplitude of a signal exceeds a digital systems ability to represent it accurately. Clipping is a type of amplitude distortion. The system reports a Clipping event when consecutive samples at the maximum value that can be represented by the digital system have been detected. Note that the maximum value that can be represented is different depending on the number of bits per sample (i.e. bits of resolution) of the audio stream. The system limits the number of reported Clipping events to typically 10 to 20 per sec. |
| Referenced | Error | CVSD HF Level Too High | Reported when a CVSD encoded audio stream is detected and there is high frequency energy above 4 kHz that is greater than -20 dBFS. |
| Referenced | Info | End of Test Event | Reported to indicate that the system has completed processing a test script for a Reference Audio file, and that the system has exited Reference Mode. This event is generated when the elapsed time from the start of test is equal to or greater than the scripted duration of a test. It is reached when the number of samples processed equals the number of samples associated with the test duration. |

Clipping

The number of consecutive samples needed to qualify as a clipping event depends on both sample rate and number of bits per sample. Table 1 specifies the number of consecutive samples at the maximum value level that will generate a Clipping event.

Table 4.25 - Clipping Event Thresholds

| Consecutive Samples | Sample Rate, Samples/sec | Resolution, bits |
|---------------------|--------------------------|------------------|
| 3 | 8000 | 16 |
| 5 | 16000 | 16 |
| 11 | 41000 | 16 |
| 2 | 64000 | 16 |

Table 4.25 - Clipping Event Thresholds (continued)

| Consecutive Samples | Sample Rate, Samples/sec | Resolution, bits |
|---------------------|--------------------------|------------------|
| 12 | 48000 | 16 |
| 24 | 96000 | 16 |

Table 4.26 - Clipping Event Thresholds

| Consecutive Samples | Sample Rate, Samples/sec | Resolution, bits |
|---------------------|--------------------------|------------------|
| 3 | 8000 | 16 |
| 5 | 16000 | 16 |
| 11 | 41000 | 16 |
| 2 | 64000 | 16 |
| 12 | 48000 | 16 |
| 24 | 96000 | 16 |

Dropout

Dropout events are reported when the average audio level (RMS) is initially above the Measurement Threshold, then falls below the Silence Threshold, and then quickly rises above the Measurement Threshold again. This approach largely disqualifies the natural inter-syllable silence and pauses that occur in natural speech, but will detect gaps caused by dropped data. Note that the system does not report dropouts that begin at very low energy levels.

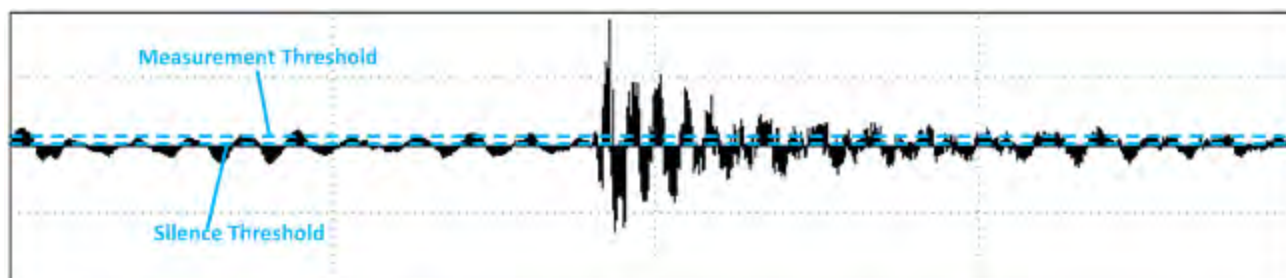


Figure 4.123 - Dropout: Measurement and Silence Threshold

Glitch

The Glitch event is reported whenever an extremely large sample to sample amplitude transition occurs that has little or no probability of occurring within natural speech or music. As illustration, back to back +N, -N, ..., +N, -N values (where N is any non-zero number), represents energy at the Nyquist frequency, or $\frac{1}{2}$ the sample rate. Neither speech nor music contain average energy levels at this frequency more than 20 dB below nominal. However, moderately large sample to sample changes in amplitude do occur, and these naturally limit how sensitive this measure can be configured.

The system uses back to back transition levels of 90 dB for music and 40 dB for speech as the threshold for reporting the Glitch event.

Such dramatic changes would typically happen only in the face of dropped samples, and serve as an additional means of detecting gross abnormalities.

4.5.5 Audio Expert System™ Window

(Click here to see an introduction video...)

This window is the working space for the Audio Expert System™. Upon opening Audio Expert System™ the window shown below will open with four main areas displayed :

- Global Toolbar - Provides play cursor controls, waveform viewing controls, and volume controls that affect all Wave Panels.
- Wave Panel - Displays the waveforms for each captured audio stream. There is a separate Wave Panel for each stream. Each panel contains local information, controls, and an event timeline specific to the displayed audio stream being shown. Other Wave Panels that may be off screen may be viewed using the vertical scroll control or by collapsing other Wave Panels.
- Event Timeline - The Event Timeline shows *Bluetooth* events, Codec events, and Audio events synchronized to the displayed waveform. There is an Event Timeline in each Wave Panel.
- Event Table – A tabular listing of *Bluetooth*, codec, and audio events with information on event severity, related *Bluetooth* frame, timestamp, and event information.

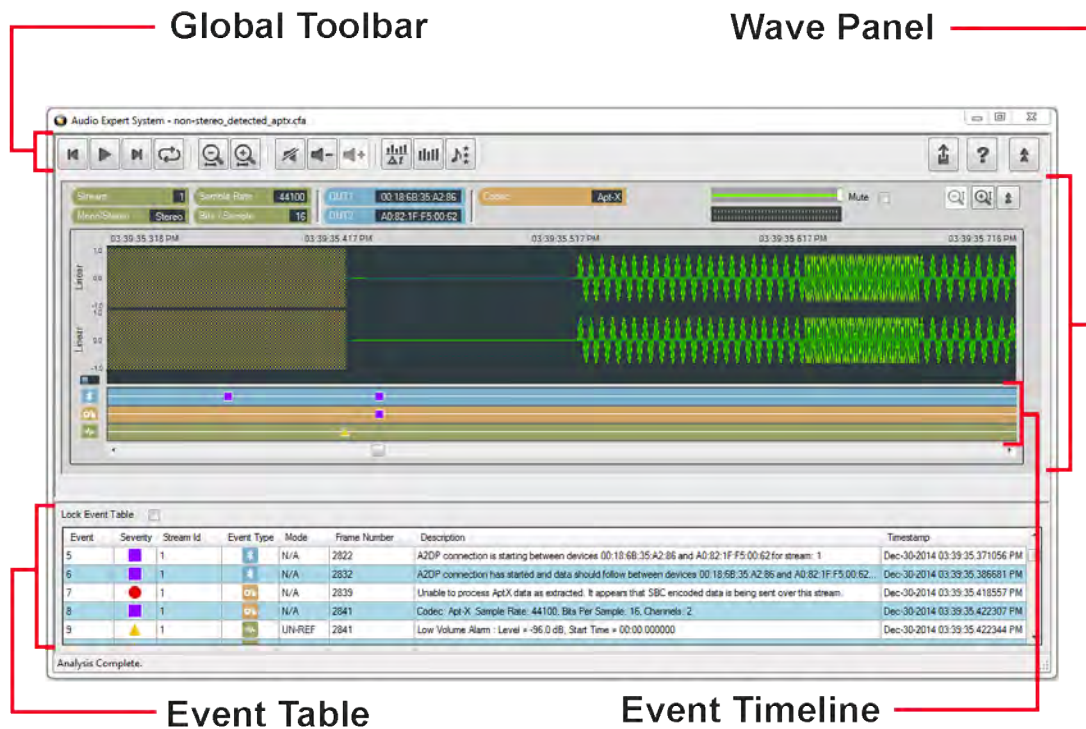








Figure 4.124 - Audio Expert System™ Window

Color Codes and Icons

The Audio Expert System™ uses standard color codes and icons to assist the user in focusing on specific issues.

Table 4.27 - Audio Expert System™ Color Codes and Icons

| Category | Sub-Category | Color Code | Icon |
|----------------|--------------|------------|---|
| Technology | Bluetooth | blue |  |
| | Codec | orange |  |
| | Audio | green |  |
| Event Severity | Information | purple |  |
| | Warning | yellow |  |
| | Error | red |  |

Note: If an Event Severity icon is surrounded by a dark line, the event is a global event and not applying to a particular captured waveform. The event is assigned to "Stream 0" in the Event Table.

The following topics describe the Global Toolbar, Wave Panel, Event Timeline and Event Table in more detail.

4.5.5.1 Global Toolbar

[\(Click here to see an introduction video...\)](#)

The global toolbar provides audio play controls, audio play cursor positioning controls, waveform viewing controls, and volume controls. Global toolbar controls apply simultaneously to all waveform panels.

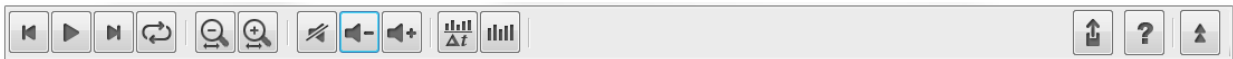


Table 4.28 - Global Toolbar Controls




| Icon | Description |
|---|---|
|  | Home: Moves play cursor to beginning of the waveform |
|  | Play : Start playing the audio from the current play cursor position. Toggles to Pause when clicked. Pause: Stops audio play back at its current position, toggles to Play when clicked. |
|  | End: Moves the play cursor to the end of the waveform |

Table 4.28 - Global Toolbar Controls (continued)













| Icon | Description |
|---|--|
|  | <p>Loop: Loops waveform playback continuously. If the Play button is visible it will toggle to the Pause. Clicking the Pause button will stop Loop playback. Clicking on the Loop button will stop the loop and the playback. If there is a selection on the waveform, only the selection will loop.</p> |
|  | <p>Horizontal Zoom Out: Increases the amount of data that is visible on the screen; however, less detail is discernible.</p> |
|  | <p>Horizontal Zoom In: Decreases the amount of data that is visible on the screen; however, more detail is discernible</p> |
|  | <p>Lock/Unlock (Operational in live mode only): Selecting Lock will freeze the waveform display; however, the Audio Expert System™ will still continue to analysis new audio data.. Selecting Unlock will jump to the waveform end and then resume following the waveform.</p> |
|  | <p>Mute: Mute will mute / unmute audio playback for all Wave Panels. Individual Wave Panel Mute control will override the Global Toolbar Mute for that panel only.</p> |
|  | <p>Volume Down: Decreases the audio playback volume of all Wave Panels based on the current volume level setting for each individual Wave Panel.</p> |
|  | <p>Volume Up: Increases the audio playback volume of all Wave Panels based on the current volume level setting for each individual Wave Panel.</p> |
|  | <p>Average Bit Rate Overlay: Displays an overlay graph of the average bit rate for the audio stream in each Wave Panel. The average is based on a 0.10 second moving window. When active, will deactivate Actual Bit Rate Overlay .</p> |
|  | <p>Actual Bit Rate Overlay: Displays an overlay graph of the instantaneous bit rate for the audio stream in each Wave Panel. When active, will deactivate Average Bit Rate Overlay</p> |
|  | <p>Export Data: Exports audio data in .raw and/or .wav format for selected Wave Panels or all the Wave Panels. This button also lets user export Event Table data in .csv format. Refer to Waveform Export Audio Data for more details .</p> |

Table 4.28 - Global Toolbar Controls (continued)

| Icon | Description |
|---|--|
|  | Help - Opens Frontline software help. |
|  | Collapse/Expand: Toggles between collapsing and expanding all Wave Panels. Note that the Wave Panel Local Controls Collapse/Expand control will locally override the Global Toolbar Collapse/Expand control. |

4.5.5.2 Wave Panel

(Click [here](#) to see an introduction video...)

The Stream Panel is where the details of the captured audio stream are presented. The Stream Panel displays the captured audio waveform along with an event timeline that displays discrete *Bluetooth*, *Codec*, and *Audio* events synchronized to the captured waveform. .

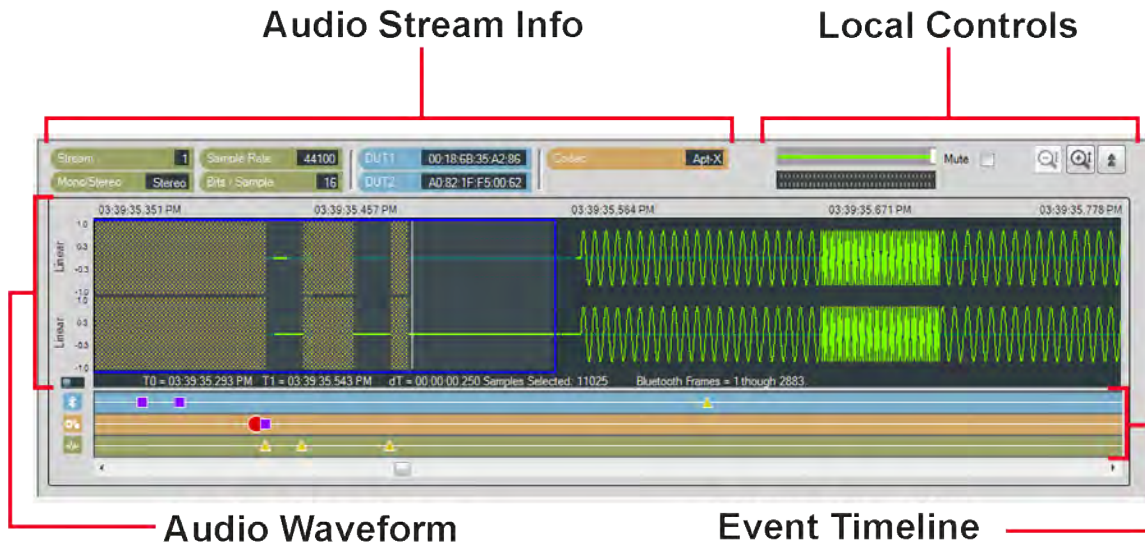


Figure 4.125 - Wave Panel

The Wave Panel contains four sections.

1. Audio Stream Info that provides users with information, such as sample rate, bit/sample, codec and DUT (Device Under Test) addresses.
2. Local Controls include audio volume controls and Indicators, "Mute", "Vertical Zoom" and "Collapse/Expand"
3. An Audio Waveform which is plotted as amplitude (linear or dB) versus time and an interactive play cursor. The play cursor appears as a white vertical line across the waveform.

- Event Timeline that shows color coded *Bluetooth* , *Codec* , and *Audio*  events. Details of these events are listed in the Audio Expert System™ Event Table.

4.5.5.2.1 Audio Stream Info

The Audio Stream Info displays Audio, *Bluetooth*, and *Codec* information (left to right in the image below) about the audio waveform displayed in the panel. This information is discovered during AVDTP signaling when the devices under test (DUT) negotiate audio streaming parameters.

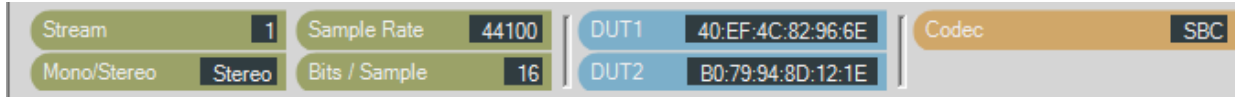


Figure 4.126 - Audio Stream Info in the Wave Panel

Table 4.29 - Audio Stream Info Tags

| Category | Name | Description |
|------------------|--------------------|--|
| Audio | Stream | A system assigned index number that represents an audio waveform between a pair of Bluetooth devices. This number appears in the Event Table for easy cross-referencing. |
| | Sample Rate | Displays the sampling frequency used to digitize the original audio. |
| | Mono/Stereo | Indicates if the audio data is monaural or stereophonic. |
| | Bits/Sample | Displays the number of bits per sample of the audio data. |
| <i>Bluetooth</i> | DUT1 | <i>Bluetooth</i> address of one device in the connection. Can be either sending or receiving the audio data. |
| | DUT2 | <i>Bluetooth</i> address of the other device in the connection. Can be either sending or receiving the audio data. |
| <i>Codec</i> | Codec | Displays the Codec type used by the captured audio stream. The supported codecs include SBC, AAC, aptX, mSBC, and CVSD. |

SBC Codec Information Pop-up

When you hover over the **Codec** tag and the Codec = SBC a pop up will appear that shows additional information about which SBC parameters can be used. The pop-up is visible as long as the cursor hovers over the **Codec** tag.

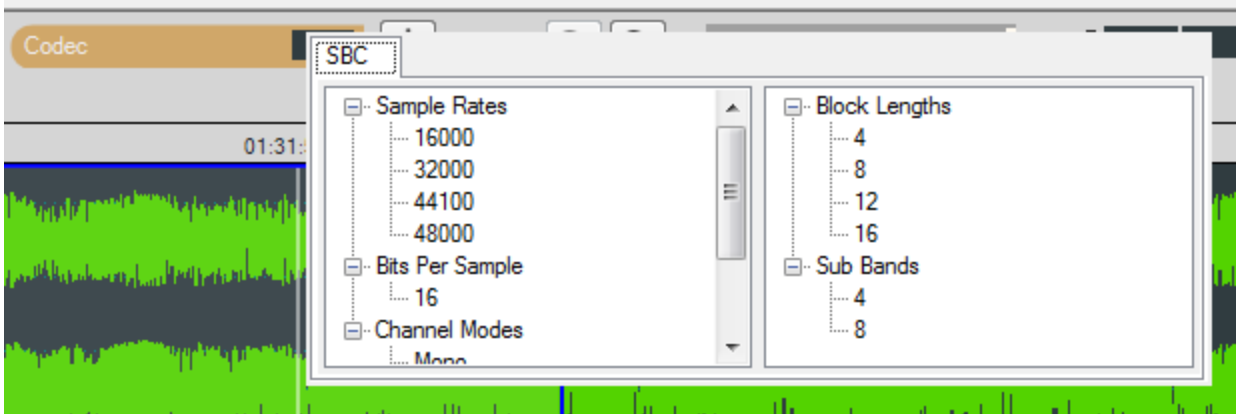


Figure 4.127 - SBC Codec Information Pop-Up on Cursor Hover Over

4.5.5.2.2 Local Controls

The Local Controls in each Wave Panel provide the user with indicators and controls for waveform display and audio play back.



Figure 4.128 - Wave Panel Local Controls

Waveform Play Back Volume



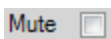
The volume slider controls the playback volume for the audio in each Wave Panel.

Audio Volume Indicator



The volume indicator shows the relative audio volume at the waveform display play cursor. When the green bars completely fill the indicator the audio volume is at its highest level. As the volume decreases, the bars will move to the right linearly, with no visible green bar indicating no audio. The volume indicator will continue to operate if the audio stream has been muted.

Mute



Checking the **Mute** check box will silence the Wave Panel's audio output. The volume indicator will respond to the audio volume but nothing will be heard. All panels can be simultaneously muted using the Audio Expert System™ Global Toolbar. The Wave Panel mute is a local control only. However, the Global Toolbar mute control will set the Stream Panel's Local Controls mute.

Vertical Zoom



Each Wave Panel contains local Vertical Zoom controls that expands or reduces the waveform display vertically. The waveform amplitude is always visible, and the Vertical Zoom controls increases or decreases the entire vertical size of the display. The vertical zoom buttons will turn gray and become inactive when the maximum and minimum values are reached.

Collapse/Expand Control



Collapse/Expand button toggles between two views. The top image indicates that the Wave Panel is expanded. When the bottom image is visible it indicates that the Wave Panel is collapsed.



When the top image is visible, clicking on it will collapse the Wave Panel to the minimum size that shows only the Stream Info and the Local Controls. When the bottom image is visible, clicking on it expands the Wave Panel to full size.

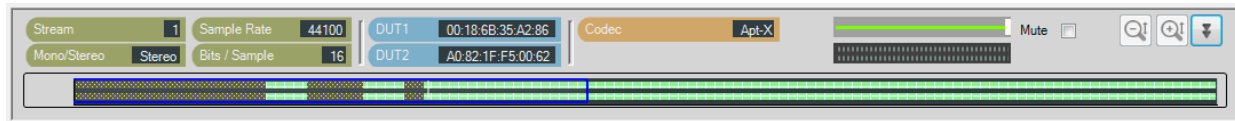


Figure 4.129 - Collapsed Wave Panel

4.5.5.2.3 Audio Waveform Panel

The Audio Waveform Panel displays the captured audio waveform. If the waveform is stereo, both channels are visible in the Wave Panel. The user can view the entire waveform or can zoom to view a portion of the waveform in more detail.

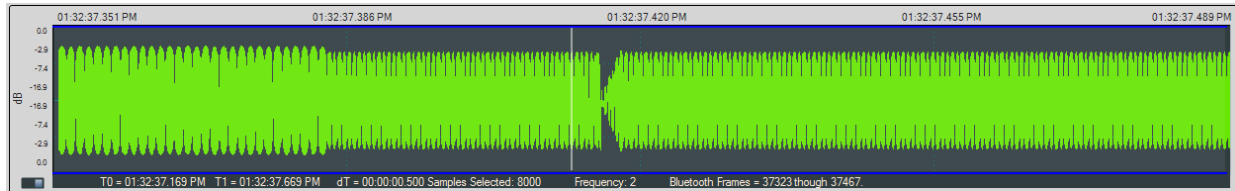
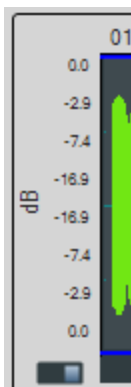


Figure 4.130 - Audio Waveform Panel in the Wave Panel

Table 4.30 - Global Toolbar Waveform Horizontal Zoom Controls

| Control | Description |
|---------|---|
| | Horizontal Zoom: Increases the amount of data that is visible on the screen; however, less detail is discernible. |
| | Horizontal Zoom: Decreases the amount of data that is visible on the screen; however, more detail is discernible. |

Waveform



The audio waveform is plotted as amplitude versus time on the Wave Panel. The amplitude scale is located on the left edge of the Wave Panel. The waveform's amplitude can be linear or in decibels. The linear range is -1.0 to +1.0. The range for the dB scale is 0 dB for the maximum positive and maximum negative values, and silence is negative infinity. A toggle switch at the bottom of the amplitude scale will switch between **Linear** scale and **dB** scale. Moving the switch to the left will display the **Linear** scale and moving it to the right will display the **dB** scale.

Play Cursor

The Play Cursor is identified by a white vertical line on the Wave Panel. The Play Cursor appears when user clicks on any point in the waveform, or, if the cursor is already present it can be dragged to another position. To drag the Play Cursor, hover the mouse cursor over the Play Cursor until the mouse cursor changes to a pointing hand; click and drag the cursor to a new position.

Waveform Segment Selection

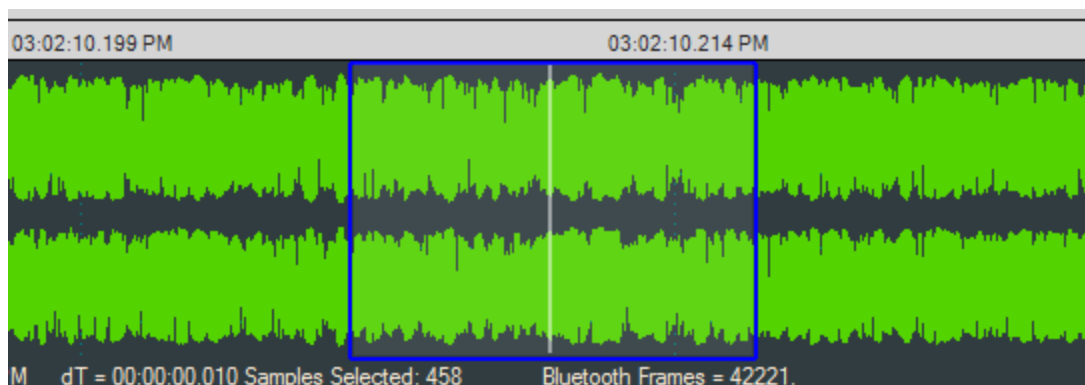


Figure 4.131 - Selection in the Audio Waveform

A waveform segment selection is identified by a blue border surrounding the selection. Procedures for selecting a segment depend on the desired actions.

Table 4.31 - Segment Selection Procedures

| Desired Action | Procedure |
|-----------------------|---|
| Loop play back | <ol style="list-style-type: none"> 1. Zoom in to the waveform segment of interest. 2. Click in the approximate center of the proposed selection. This will place the Play Cursor in the area to be selected. 3. Move the mouse cursor to the right or left of the Play Cursor, click and hold, then drag over the waveform segment of interest. Release the mouse key. The selection is surrounded by a blue border. |
| View waveform details | <ol style="list-style-type: none"> 1. Zoom in to the segment of interest. 2. Move the mouse cursor to the right or left limit of the waveform segment of interest; click and hold, then drag over the waveform segment of interest. Release the mouse key. The selection is surrounded by a blue border. |

For either of the procedures described in the table above, once the selection is made details of the segment appear below and to the left of the waveform. These details include selection start and stop range ("T0" and "T1"), the time difference ("dT"), samples selected, frequency, and "Bluetooth Frames" selected.

Right-clicking in the Waveform panel will open a pop up menu (see [Wave Panel & Event Table Pop-up Menu on page 246](#)). Selecting **Zoom to Selection** will expand the selection to the full width of the Wave Panel. Other selection option in the pop up are **Select Area**, **Clear Selection**, and **Copy Selection**.

Actual Bitrate Overlay Display

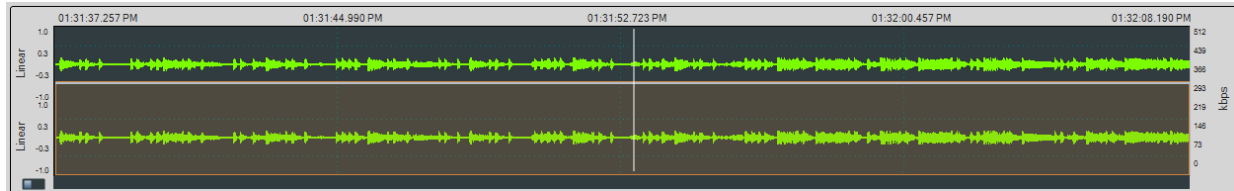




Figure 4.132 - Actual Bitrate Overlay

The Average and Actual audio stream bitrate graphs can be displayed over the audio waveform using the Global Toolbar Average Bitrate Overlay  and Actual Bitrate Overlay  buttons respectively. These are presented as overlays onto the main Wave Panel so the user can correlate audio issues with bitrate changes and the like. The scale is in kbps (kilo bits per second). Hovering over the bitrate scale will display a pop-up showing the bitrate at the play cursor position.

Actual Bitrate is based on the throughput at the Codec level.

The Average Bitrate is the moving average over 0.1 sliding-second window.

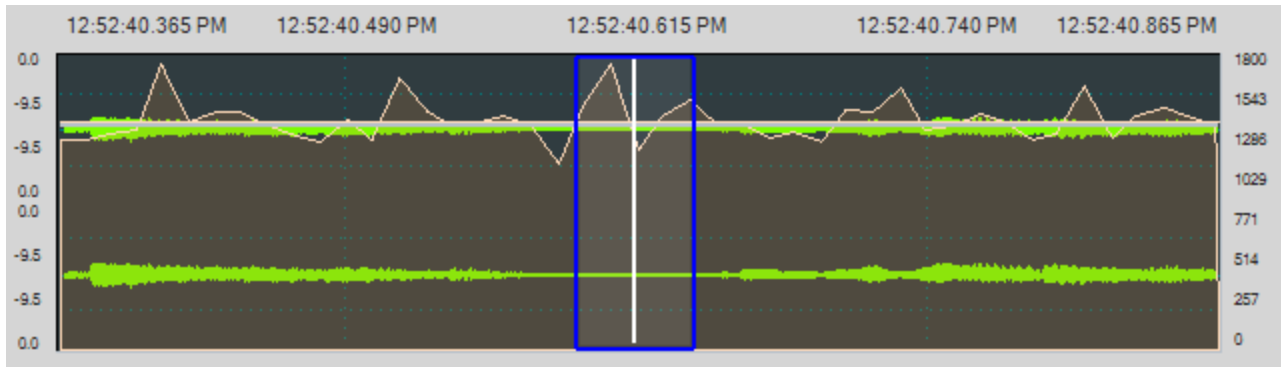



Figure 4.133 - Average Bitrate Overlay

All of the information for calculating the Actual and Average Bitrate is in the codec data frame header.

4.5.5.2.4 Event Timeline

[\(Click here to see an introduction video...\)](#)

The Event Timeline in the Wave Panel shows the *Bluetooth* , *Codec* , and *Audio*  events related to the waveform being viewed. The events are synchronized in time to the waveform displayed in the Wave Panel. The event severity is displayed as Information , Warning , and Error .

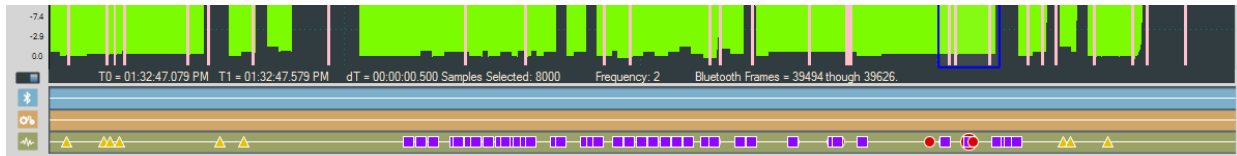


Figure 4.134 - Event Timeline Shown with Wave Panel

Clicking on an event in the Event Timeline shows a relevant selection in the Audio Waveform Panel. The size of the selection depends on the number of frames associated with the selected event. This selection will appear in all Wave Panels; however, the event severity icon will only appear in the Wave Panel associated with the event.

To assist the user with viewing events in detail, the Event Timeline will zoom in and out in sync with the Wave Panel.

Event Timeline Example

This example shows that event 159 was selected in the Event Table resulting in the severity icon being enlarged in the Event Timeline. The system automatically selected the surrounding area—the blue outline.

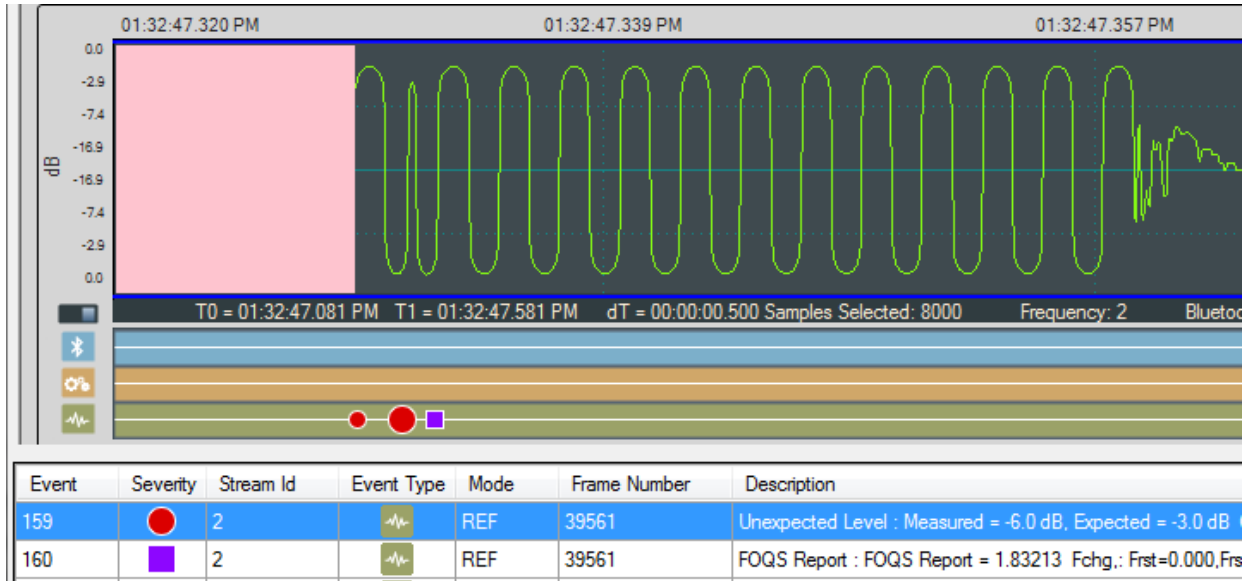


Figure 4.135 - Example: Event Table Selection Shown in Event Timeline

Event Pop Up

When the cursor hovers over a selected event severity icon in the Event Timeline, a pop-up will display the event class, severity, and associated *Bluetooth* frame.

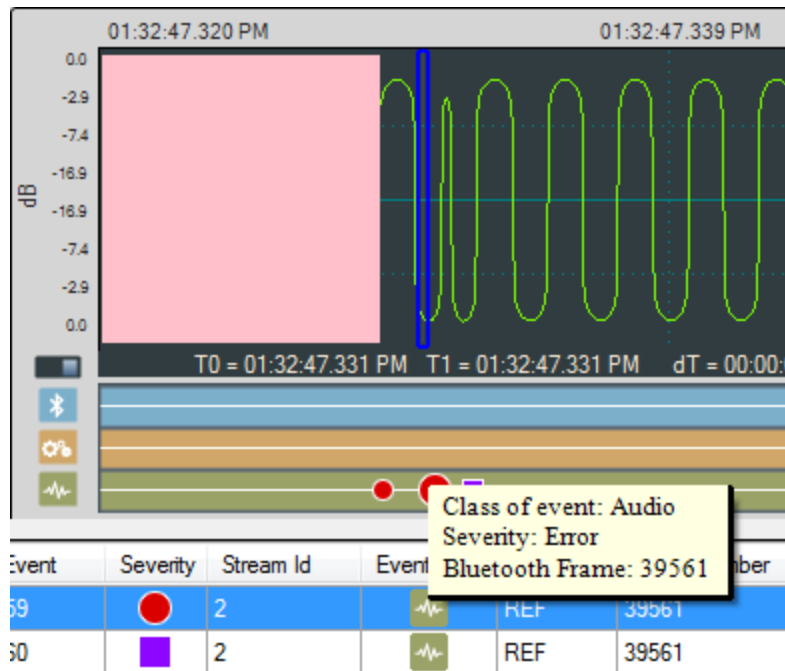


Figure 4.136 - Event Timeline Selected Event Pop Up

4.5.5.3 Event Table

[\(Click here to see an introduction video...\)](#)

The Event Table lists all audio stream events . Clicking on an event will select that event in the Event Timeline in the Wave Panel. If the selected event is outside the visible area of the waveform, the waveform will move and bring the selected event to the center of the display. The event icon in the Event Timeline is also centered and the selected icon will be larger than the non-selected event icons. Selecting one or more events in the table will highlight the associated frames in the standard Frontline software windows, such as **Frame Display**, **Coexistence View**, **Bluetooth Timeline**, etc. .

| Event | Severity | Stream Id | Event Type | Mode | Frame Number | Description | Timestamp |
|-------|----------|-----------|------------|------|--------------|---|--------------------------------|
| 17 | ▲ | 1 | 🔗 | N/A | 3039 | Packet retransmission. | Mar-31-2014 12:52:38.080991 PM |
| 18 | ■ | 1 | 🔗 | N/A | 4094 | A2DP paused between devices 98:0D:2E:23:B6:2E and 00:07:62:0F:00:00 for stream: 1 using codec: SBC | Mar-31-2014 12:52:45.553569 PM |
| 19 | ■ | 1 | 🔗 | N/A | 4095 | A2DP paused between devices 98:0D:2E:23:B6:2E and 00:07:62:0F:00:00 for stream: 1 using codec: SBC | Mar-31-2014 12:52:45.617944 PM |
| 20 | ▲ | 0 | 🔗 | N/A | 4101 | SCO connection request. | Mar-31-2014 12:52:46.151071 PM |
| 21 | ■ | 2 | 🔗 | N/A | 4105 | SCO connection established between devices 98:0D:2E:23:B6:2E and 00:07:62:0F:00:00 for stream: 2 using cod... | Mar-31-2014 12:52:46.504191 PM |
| 22 | ■ | 3 | 🔗 | N/A | 4105 | SCO connection established between devices 00:07:62:0F:00:00 and 98:0D:2E:23:B6:2E for stream: 3 using cod... | Mar-31-2014 12:52:46.504191 PM |
| 23 | ■ | 2 | 🔗 | N/A | 4108 | Codec: CVSD Frequency: 64000, Bits Per Sample: 16, Channels: 1 | Mar-31-2014 12:52:46.806067 PM |
| 24 | ■ | 3 | 🔗 | N/A | 4256 | Codec: CVSD Frequency: 64000, Bits Per Sample: 16, Channels: 1 | Mar-31-2014 12:52:47.357946 PM |
| 25 | ■ | 2 | 🔗 | N/A | 13222 | SCO disconnected between devices 98:0D:2E:23:B6:2E and 00:07:62:0F:00:00 for stream: 2 using codec: CVSD | Mar-31-2014 12:53:04.151789 PM |
| 26 | ■ | 3 | 🔗 | N/A | 13222 | SCO disconnected between devices 00:07:62:0F:00:00 and 98:0D:2E:23:B6:2E for stream: 3 using codec: CVSD | Mar-31-2014 12:53:04.151789 PM |
| 27 | ■ | 1 | 🔗 | N/A | 13253 | A2DP resumed between devices 98:0D:2E:23:B6:2E and 00:07:62:0F:00:00 for stream: 1 using codec: SBC | Mar-31-2014 12:53:05.446738 PM |
| 28 | ■ | 1 | 🔗 | N/A | 13254 | A2DP resumed between devices 98:0D:2E:23:B6:2E and 00:07:62:0F:00:00 for stream: 1 using codec: SBC | Mar-31-2014 12:53:05.474864 PM |
| 29 | ▲ | 0 | 🔗 | N/A | 13479 | Packet retransmission for unknown CID. | Mar-31-2014 12:53:07.712976 PM |
| 30 | ▲ | 1 | 🔗 | N/A | 14187 | AVDTP packet loss detected based on missing packet sequence number. | Mar-31-2014 12:53:13.742943 PM |
| 31 | ▲ | 1 | 🔗 | N/A | 14351 | AVDTP packet loss detected based on missing packet sequence number. | Mar-31-2014 12:53:15.385434 PM |

Figure 4.137 - Event Table

Several events can be selected by clicking and dragging over the events, or by holding down the Shift key and clicking on events. To select events that are not adjacent hold down the Ctrl key and click on the events.




When selecting multiple events, the Wave Panels will not scroll to the selected events.

The Event Table contains eight columns.

Table 4.32 - Event Table Columns

| Name | Value | Description |
|------------------|---------|--|
| Event | integer | System generated sequential numbering of events. |
| Severity | | Information - provides information of interest but does not indicate a problem event. |
| | | Warning - identifies a potential problem where further investigation may be appropriate |
| | | Error - identifies a definite problem. |
| Stream Id | integer | A system generated ID that is assigned in the order that the audio streams are detected. The ID is not maintained between captures for the same device with the same audio. It identifies the Wave Panel where the event can be viewed. The ID appears in the Audio Stream Info of the Wave Panel. |

Table 4.32 - Event Table Columns (continued)

| Name | Value | Description |
|--------------|---|---|
| Event Type |  | Bluetooth -Events generated by analyzing Bluetooth protocol activities. |
| |  | Codec -Events generated from analyzing the audio coding/decoding activities. |
| |  | Audio -Events generated by analyzing the audio data. |
| Mode | N/A | Mode does not apply to this event. |
| | REF | Referenced Mode. Refer to 4.5.3.2 Referenced Mode on page 217 . |
| | UN-REF | Non-Referenced Mode. Refer to 4.5.3.1 Non-Referenced Mode on page 217 . |
| Frame Number | integer | The system generated identification for a specific frame. |
| Description | | Details and explanation about this event. |
| Timestamp | clock date and time | A system generated time stamp for each frame. |

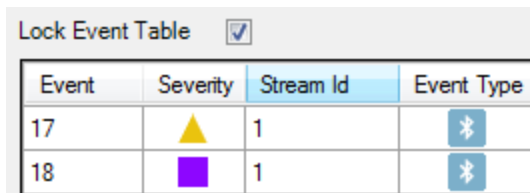
Sorting

Event table entries are sortable by column. Left-click on the column heading to sort.

Event Table Pop-Up Menu

Right-clicking with the cursor over the Event Table will open a menu of additional options. For more on this option see [Wave Panel & Event Table Pop-up Menu on page 246](#).

Lock Event Table

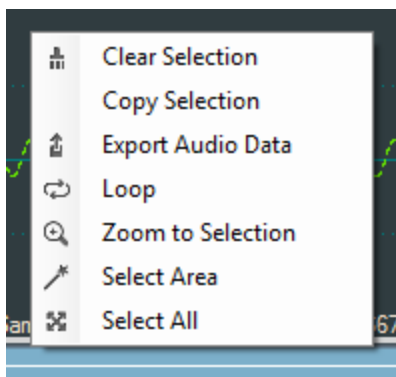


The **Lock Event Table** checkbox is available in live mode only. Clicking to check the box will prevent the Event Table from scrolling during live capture. Un-checking the box will resume scrolling of events as they are detected. When analyzing a capture file the checkbox has no effect.

4.5.5.4 Wave Panel & Event Table Pop-up Menu

Additional Wave Panel and Event Table options are available by right clicking the mouse with the cursor anywhere in the Wave Panel or in the Event Table.

Wave Panel Pop-up Menu Actions

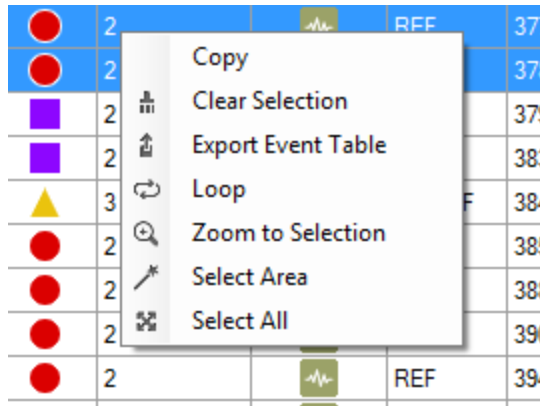


Right-clicking anywhere in the Wave Panel will provide you with a selection of the following actions.

Table 4.33 - Wave Panel Pop-up Menu Selections

| Option | Description |
|-------------------|---|
| Clear Selection | Clears the current selection in the viewer |
| Copy Selection | Saves a copy of the selection to the computer clipboard. The clipboard can be pasted into a Word document, an e-mail, or other Windows clipboard-compatible application. |
| Export Audio Data | Opens the Export pop-up menu with options to export the waveform as a .raw, .wav, or Event Data. For additional details on exporting refer to Waveform Display Export . |
| Loop | Loops through the audio selected on the Wave Panel. |
| Zoom to Selection | Expands or compresses the selection to fill the Wave Panel view. |
| Select Area | When the mouse cursor is positioned over data (not fill, pause, or gaps) in the Wave Panel and selecting this option will select all the data between and fills, pauses, or gaps. |
| Select All | Selects the entire waveform |

Event Table Pop-up Menu Actions




Right-clicking in the Event Table will provide you with a selection of the following actions.

Table 4.34 - Event Table Pop-up Menu Selection

| Options | Description |
|--------------------|---|
| Copy | Copies the selected events to Windows clipboard as text. |
| Clear Selection | Clears the current event selection in the table |
| Export Event Table | Copies the current event selection and saves it as a .csv file. For additional details on exporting refer to Event Table Export . |
| Loop | Loops through the audio selected on the Wave Panel. |
| Zoom to Selection | Expands the Event Table selection to fill the Wave Panel view. |
| Select Area | Expands the selection. |
| Select All | Selects all events. |

4.5.5.5 Export Audio Data

There are two ways to export audio data:

1. Clicking the Audio Expert System™ window **Global Toolbar** Export button .
2. Right-click in a Stream Panel Wave Panel and a pop-up menu will appear. Select **Export**.

Two windows will appear:

1. The standard Windows Save As.
2. The **Export Audio Data** dialog.

In the Windows Save As window enter a **File name** and directory location. Click on **Save**.

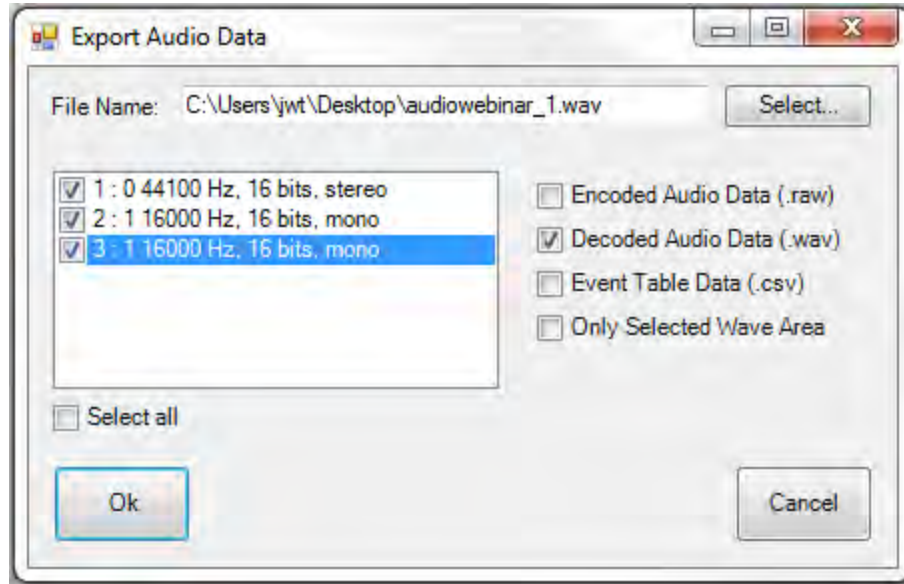


Figure 4.138 - Export Audio Data dialog

The Save As window will close, and the file name will appear in the **File Name** field in the **Export Audio Data** window. Should the file name need to be changed, click on the **Select** button and the Windows Save As dialog will open. By default the .wav file extension is used in the file name.







In the window below **File Name** will appear a list of **Stream IDs** with a description from the Audio Stream Info . If opening from the Audio Expert System™ **Global Toolbar** all **Stream IDs** are checked by default. If opening from a Wave Panel, the **Stream ID** where the export dialog was opened is automatically checked. You can check each stream that is to be exported. For convenience checking **Select all** below the stream list window will place checks in all streams.

Export Options

After selecting the streams to export, select the desired formats to export.

Table 4.35 - Export Audio Data Format Options

| Option | Description |
|-------------------------|--|
| Encoded Audio Data | Exports the selected files as .raw format. The audio data is in an encrypted format and user will need a codec to decode it. |
| Decoded Audio Data | Exports the selected files as .wav format that can be played on a wide variety of media players. |
| Event Table Data | Exports a text .csv file of all the detected events |
| Only Selected Wave Area | Exports the Encoded, Decoded, or Event Data for the selected waveform. This option is only active if a selection has been made in one of the Wave Panels |

| | | | |
|---|--------------------|-----------|----------------------|
|  | audiowebinar_1.csv | 39 KB | Microsoft Excel C... |
|  | audiowebinar_1.raw | 5,439 KB | RAW File |
|  | audiowebinar_1.wav | 38,652 KB | Wave Sound |
|  | audiowebinar_2.csv | 39 KB | Microsoft Excel C... |
|  | audiowebinar_2.raw | 299 KB | RAW File |
|  | audiowebinar_2.wav | 7,227 KB | Wave Sound |

Click on **OK** to save the waveform. The dialog will close and a series of progress bars will appear. Each progress bar is associated with a file for each export option. The exported files will have the following syntax: `<filename>_n.<filetype>`, where `<filename>` = the name entered into the File Name field, `n` = the stream id number (1, 2, 3, ...), and `<filetype>` = "raw", "wav", and "csv". The image shows an example where the user exported **Stream**

Id's 1 and 2 in Encoded Audio , Decoded Audio , and Event Table data to filename "audiowebinar".

Click on **Cancel** to close the window without exporting.

4.5.5.6 Export Event Table

Right-clicking in the Event table will open a pop-up menu with the option to **Export Event Table**. This option will export selected events in the in comma separated variable (.csv) format for used in Microsoft Excel or any other Windows .csv compatible application.

First select the events to export. Multiple events are selectable by selecting an event then holding the Shift key while clicking on another event. This will select all events between the two selections. If the selections are not adjacent you can hold the Ctrl (control) key while clicking events.

Next right-click anywhere in the Event Table to open the pop-up menu and click on the **Export Event Table** option. A Windows **Save As** dialog will open. Enter a file name and select a file location and click on **Save**. A confirmation dialog will open. Click **OK** to close the confirmation dialog.

If you have not selected an event in the table before exporting, a warning to "Please select an event row first." appears.

4.5.6 Frame, Packet, and Protocol Analysis Synchronization

The Audio Expert System™ module integrates seamlessly with Frontline software with common timestamping of *Bluetooth* protocol data, audio events, audio waveform display, and codec events. The audio expert data and results are synchronized and coordinated with the existing Frontline software data views, such as **Frame Display**, **Bluetooth Timeline**, etc. to expedite the root-cause analysis of *Bluetooth* protocol related audio issues. When a frame is selected in any Frontline software data views, the corresponding audio data associated with those frames is also selected in the Wave Panel, Event Timeline and Event Table and vice-verse.

Protocol analysis tools synchronized to the Audio Expert System™ include:

- **Frame Display**
- **Coexistence View**
- **Bluetooth Timeline**
- **Message Sequence Chart**
- **Packet Error Rate Statistics**

When a portion of the waveform is selected in the Wave Panel, all frames within the selection will be highlighted in the **Frame Display**, **Coexistence View**, and **Bluetooth Timeline**.

Note: If the **Frame Display** is filtered to show non-audio events then the frames associated with selected audio events may not show.

4.6 Bluetooth Protocol Expert System



Bluetooth Protocol Expert System

The *Bluetooth* Protocol Expert System is used to debug protocol-related events for *Bluetooth* protocols. The expert system provides the ability to interactively select protocol events from a table of events in live capture mode or in analyzing a previously captured file. Selecting an event will

dynamically link the related packet selection to the ComProbe software **Frame Display**, **Coexistence View**, **Message Sequence Chart**, **Bluetooth Timeline**, and **Packet Error Rate Statistics (PER Stats)**.

Protocol error events appearing in the **Protocol Events** pane identify the related *Bluetooth* specification reference that is likely to point to a solution to the error. The expert system references *Bluetooth* specification 5.0 and the following protocols for both Classic *Bluetooth* and *Bluetooth* low energy.

- L2CAP
- A2DP
- SDP
- SMP
- ATT


4.6.1 Starting the *Bluetooth* Protocol Expert System


To use the *Bluetooth* Protocol Expert System the user must have

- Current Premium Maintenance
- Set in Target hardware with *Bluetooth* Protocol Expert System license installed and connected to the PC. This is a requirement for both live capture and when viewing a saved capture file.

For live capture, set up the Set in Target device datasource and begin capturing data.

For viewing a capture file, load the saved file from the **Control** window **File** menu.

Bluetooth Protocol Expert System Window is opened by clicking on  on the **Control** window toolbar. If the

Set in Target hardware is not licensed for *Bluetooth* Protocol Expert System, a tooltip will appear with "Bluetooth Protocol Expert System is not licensed. Please contact sales@fte.com." Click on the  or select **Bluetooth**

Protocol Expert from the **View** menu. The *Bluetooth* Protocol Expert System window will open.



When the protocol analyzer begins analysis of the captured data, the **Bluetooth Protocol Expert System** window status bar (bottom of the window) will show **Progress** with a progress bar. The expert system will search and evaluate for protocol events for warnings and errors. When the expert system has completed, the status bar will show **Processing stopped**. If no protocol warnings or errors are detected, the window will remain empty of data.

4.6.2 Bluetooth Protocol Expert System Window


This window is the working space for the *Bluetooth* Protocol Expert System. Upon opening *Bluetooth* Protocol Expert System by clicking on the **Control** window  button, the window shown below will open with four main areas displayed described in the table below. Detailed explanations of each window section follow.

Table 4.36 - *Bluetooth* Protocol Expert System Window Panes

| Section | Description |
|---------------------------------|--|
| Connections | Displays the <i>Bluetooth</i> master and slave device connections with associated link layer logic transport type. |
| Statistics | Displays the protocol statistics associated with the warning or error selected in the Protocol Events pane, or associated with the selected <i>Bluetooth</i> address and protocols selected in Connections pane. Tabbed sections contain the statistics for the protocols associated with the analyzed data. Statistics will vary depending on the protocol. |
| Protocol Events | Displays the <i>Bluetooth</i> protocol warnings and errors. Clicking on an event will select the associated protocol tab in the Statistics pane. |

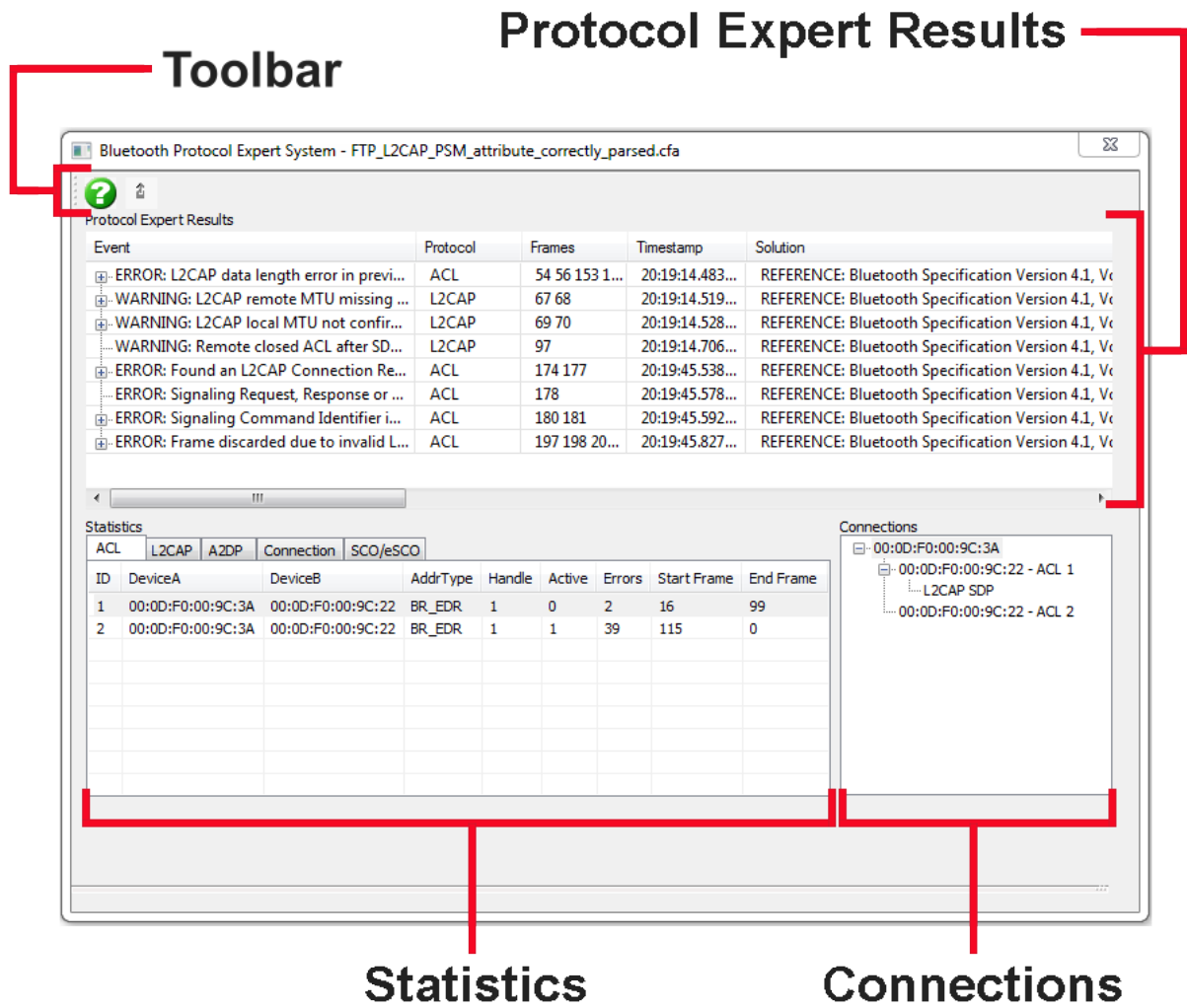


Figure 4.139 - Bluetooth Protocol Expert System Window

4.6.2.1 Expert System Connections Pane

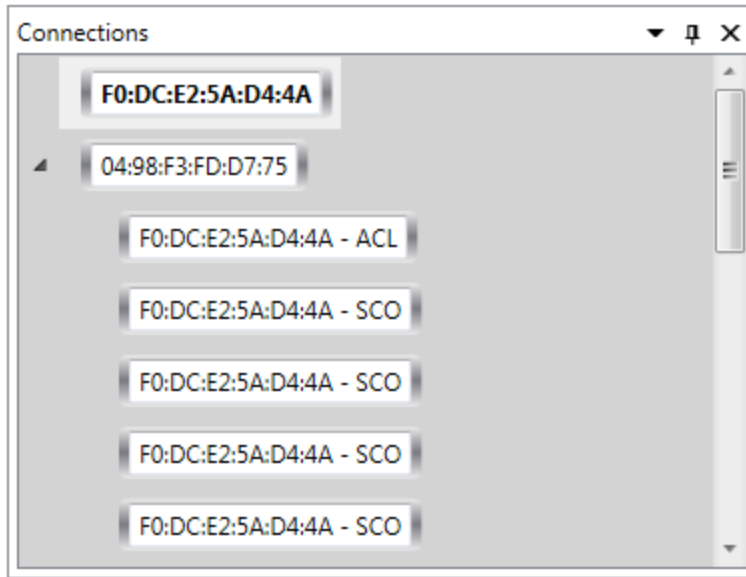


Figure 4.140 - Bluetooth Protocol Expert System **Connections** Pane

The **Connections** pane provides a chart of all the connected devices from the current live recording session or from a loaded capture file that have a protocol error or warning appearing in the **Protocol Events** pane. Devices are identified by their BD_ADDR. A device address with an arrow symbol will expand to show the connected device and the link layer logical transport type.

4.6.2.2 Expert System Statistics Pane

The Statistics pane contains detailed information about the links, protocols, and connections associated with frames or range of frames and devices of detected events. The tabs across the top list the links, protocols, or connections. Details are contained with the

| Id | Source CCID | Destination CCID | Extra Features | Mode | Transmit MTU | Receive MTU | Local PSM | Remote PSM | Data Transmitted | Data Received | Transmit Mps |
|----|-------------|------------------|----------------|------|--------------|-------------|-----------|------------|------------------|---------------|--------------|
| 1 | 40 | 0 | "" | "" | 668 | 256 | 0 | 0 | 0 | 0 | 0 |
| 2 | 41 | 0 | "" | "" | 1013 | 2048 | 0 | 0 | 0 | 0 | 0 |
| 3 | 40 | 0 | "" | "" | 668 | 256 | 0 | 0 | 0 | 0 | 0 |
| 4 | 40 | 0 | "" | "" | 668 | 256 | 0 | 0 | 0 | 0 | 0 |
| 5 | 40 | 0 | "" | "" | 668 | 256 | 0 | 0 | 0 | 0 | 0 |
| 6 | 40 | 0 | "" | "" | 668 | 256 | 1 | 0 | 0 | 0 | 0 |
| 7 | 40 | 0 | "" | "" | 668 | 256 | 1 | 0 | 0 | 0 | 0 |

Figure 4.141 - Bluetooth Protocol Expert System **Statistics** Pane

Table 4.37 - Bluetooth Protocol Expert System Statistics Pane

| Tab | Tab Description | Column | Column Description |
|------------|--|-----------------|---|
| ACL | An asynchronous (packet switched) connection between devices created on LMP level. | ID | System assigned identifier for ACL connections. |
| | | Device A | Contains the BD_Addr of a device in the connection. |
| | | Device B | Contains the BD_Addr of a device in the connection. |
| | | AddrType | BR_EDR or LE |
| | | Handle | |
| | | Active | |
| | | Errors | |

Table 4.37 - Bluetooth Protocol Expert System Statistics Pane (Continued)

| Tab | Tab Description | Column | Column Description |
|--------------------|--|----------------------------------|---|
| L2CAP | L2CAP provides connection-oriented and connectionless data services to upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions. | ID | System assigned identifier for ACL connections |
| | | Source CID | Channel Identifier for the source device. |
| | | Destination CID | Channel Identifier for the destination device. |
| | | Extra Features | |
| | | Mode | |
| | | Transmit MTU | Maximum Transmission Unit in bytes during transmission. |
| | | Receive MTU | Maximum Transmission Unit in bytes during receive. |
| | | Local PSM | Local device Protocol and Service Multiplexer. |
| | | Remote PSM | Remote device Protocol and Service Multiplexer. |
| | | Data Transmitted | |
| | | Data Received | |
| | | Transmit Mps | |
| | | Receive Mps | |
| | | Transmit Window | |
| | | Receive Window | |
| | | Number of Retransmissions | |
| Active | | | |
| Error Count | Number of errors associated with this L2CAP Id . | | |
| A2DP | | | |
| Connection | Provides a cross-reference among the Statistics tab Id , the Frame Display frame range, and the Connection Type . | Id | System assigned identification |
| | | Connection Type | The protocol type |
| | | Parent Id | ???????? |
| | | Statistics Id | The system assigned Id that appears in the Statistics Connection Type tab. |

Table 4.37 - Bluetooth Protocol Expert System Statistics Pane (Continued)

| Tab | Tab Description | Column | Column Description |
|-----------------|--|--------------------|--|
| SCO/eSCO | Synchronous Connection-oriented (SCO)/extended SCO. | Id | System assigned identification. |
| | | Type | SCO or eSCO |
| | | Air Mode | Part of the <i>voice_settings</i> parameter in the air mode negotiations designed to improve or optimize audio quality during transmissions. SCO: CVSD, A-law, μ -law. eSCO: CVSD, A-law, μ -law, transparent. |
| | | Handle | |
| | | Active | |
| | | Error Count | |
| Device | This tab serves the purpose of assigning a unique expert system identification to the devices listed in the Connections pane. | Id | System assigned identification. |
| | | Address | BD_ADDR of a device found in the Connections pane. |

Any column in the Protocol Events list can be sorted in ascending or descending order. Refer to [Expert System Table Sorting on page 259](#) for sorting instructions.

4.6.2.3 Expert System Protocol Events Pane

| Event | Severity | Event Text | Protocol | Frame Id | Solution | Time |
|-------|----------|-------------------------------|----------|----------|---|--------------------------------|
| 1 | ● | Unable to negotiate L2CAP lir | L2CAP | 1383 | REFERENCE: Bluetooth Specification Version 5.0, Volume 3, | May-29-2015 01:46:13.043601 PM |
| 2 | ● | Unable to negotiate L2CAP lir | L2CAP | 1521 | REFERENCE: Bluetooth Specification Version 5.0, Volume 3, | May-29-2015 01:46:14.484855 PM |
| 3 | ● | Unable to negotiate L2CAP lir | L2CAP | 1561 | REFERENCE: Bluetooth Specification Version 5.0, Volume 3, | May-29-2015 01:46:14.973606 PM |
| 4 | ● | Unable to negotiate L2CAP lir | L2CAP | 1621 | REFERENCE: Bluetooth Specification Version 5.0, Volume 3, | May-29-2015 01:46:16.493610 PM |
| 5 | ● | Invalid SCO connection paran | | 11207 | REFERENCE: Bluetooth Specification Version 4.1, Volume 2, | May-29-2015 01:47:40.391315 PM |

Bluetooth protocol events that generate a warning or an error in the expert system are listed in the **Protocol Events** pane. Events are listed in the order that they occur.

Table 4.38 - Protocol Events Pane Fields

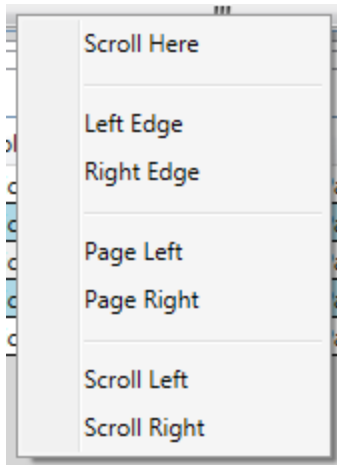
| Row Field | Description |
|--------------|--|
| Event | System assigned event number. Events are numbered in the order that they appear. |

Table 4.38 - Protocol Events Pane Fields (Continued)

| Row Field | Description |
|-------------------|---|
| Severity | <p>▲ = Warning. The event has not created a failure, but should receive some attention and further investigation..</p> <p>● = Error. The event has identified a situation that does not conform to the <i>Bluetooth</i> specification. Corrective action is required.</p> |
| Event Text | Event description. |
| Protocol | Protocol in which the event occurred. |
| Frame Id | Frame where the event occurred. Clicking in the event row will select the related Statistics pane protocol tab and protocol Id . The corresponding frame is selected in the Frame Display , Event Display , Message Sequence Chart , Coexistence View , and Bluetooth Timeline or Bluetooth low energy Timeline . |
| Solution | A solution to the event is provided by reference to the Bluetooth specification that applies to the Event Text content. |
| Time | Event timestamp. |

Any column in the Protocol Events list can be sorted in ascending or descending order. Refer to [Expert System Table Sorting on page 259](#) for sorting instructions.

4.6.2.4 Expert System Window Scroll Bar Navigation



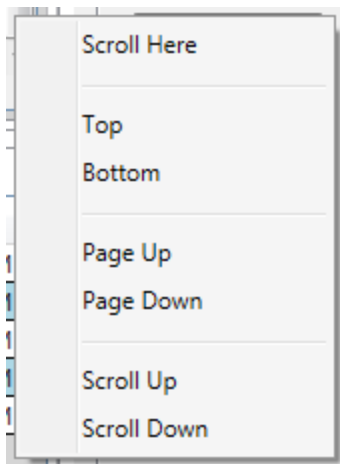
Some tabs in the Statistics pane display a horizontal scroll bar that can be clicked and dragged to view the tab columns. An alternative scroll navigation is to right-click the mouse cursor on the scroll bar. A navigation menu will appear, and you click on the direction and amount of scrolling to move the horizontal scroll bar in discrete steps.

Table 4.39 - Horizontal Scroll Bar Navigation Sections

| Selection | Description |
|-------------|--|
| Scroll Here | Scrolls to the point on the scroll pane where the mouse was last positioned. |

Table 4.39 - Horizontal Scroll Bar Navigation Sections (Continued)

| Selection | Description |
|-------------------|---|
| Left/Right Edge | Scrolls the table to the beginning (left edge) or to the end (right edge) |
| Page Left/Right | Left: Moves the current right edge to the left edge of the current view range. Right: Moves the current left edge to the right edge of the current view range. |
| Scroll Left/Right | Moves the table in small increments to the left or right. Same action as the left/right scroll arrows at the ends of the scroll bar. |



Some tabs in the Statistics pane display a vertical scroll bar that can be clicked and dragged to view the tab columns. An alternative scroll navigation is to right-click the mouse cursor on the vertical scroll bar. A navigation menu will appear, and you click on the direction and amount of scrolling to move the scroll bar vertically in discrete steps.

Table 4.40 - Vertical Scroll Bar Navigation Sections

| Selection | Description |
|----------------|--|
| Scroll Here | Scrolls to the point on the scroll pane where the mouse was last positioned. |
| Top/Bottom | Scrolls the table to the first row (top) or to the last row (bottom) |
| Page Up/Down | Up: moves the current view bottom row to the top row of the current view range. Down: Moves the current view top row to the bottom row of the current view range. |
| Scroll Up/Down | Moves the table one row up or down. |

4.6.2.5 Expert System Table Sorting

Tables in the Bluetooth Protocol Expert System can be sorted in ascending or descending order. This process includes tables in the **Statistics** pane and the **Protocol Events** pane.

1. In any table click in the header for the column you want to sort. The column header will turn blue and an arrow head will appear.
2. If the arrow head is pointing up, the column is sorted in ascending order. If the arrow head is pointing down the column is sorted in descending order.
3. To change the direction of the sort, click in the column header to change the arrow head direction accordingly.

All other columns in the table are sorted relative to the selected column sort. Refer to the following Statistics pane images for an example.

| Id | Type | Air Mode | Handle | Active | Error Count |
|----|------|-------------|--------|--------|-------------|
| 1 | eSCO | Transparent | 1 | False | 0 |
| 2 | SCO | CVSD | 1 | False | 0 |
| 3 | eSCO | Transparent | 1 | False | 0 |
| 4 | SCO | CVSD | 1 | False | 0 |

Figure 4.142 - Sorting **Id** Ascending

| Id | Type | Air Mode | Handle | Active | Error Count |
|----|------|-------------|--------|--------|-------------|
| 4 | SCO | CVSD | 1 | False | 0 |
| 2 | SCO | CVSD | 1 | False | 0 |
| 1 | eSCO | Transparent | 1 | False | 0 |
| 3 | eSCO | Transparent | 1 | False | 0 |


Figure 4.143 - Sorting **Air Mode** Ascending; Note **Id** Sort

| Statistics | | | | | | |
|------------|-------|-------------|------------|----------|-------------|--|
| ACL | L2CAP | A2DP | Connection | SCO/eSCO | Device | |
| Id | Type | Air Mode | Handle | Active | Error Count | |
| 3 | eSCO | Transparent | 1 | False | 0 | |
| 1 | eSCO | Transparent | 1 | False | 0 | |
| 4 | SCO | CVSD | 1 | False | 0 | |
| 2 | SCO | CVSD | 1 | False | 0 | |

Figure 4.144 - Sorting **Air Mode** Descending; Note how other columns follow.

4.7 Analyzing Byte Level Data

4.7.1 Event Display

To open this window click the **Event Display** icon  on the **Control** window toolbar.

The **Event Display** window provides detailed information about every captured event. Events include data bytes, data related information such as start-of-frame and end-of-frame flags, and the analyzer information, such as when the data capture was paused. Data bytes are displayed in hex on the left side of the window, with the corresponding ASCII character on the right.

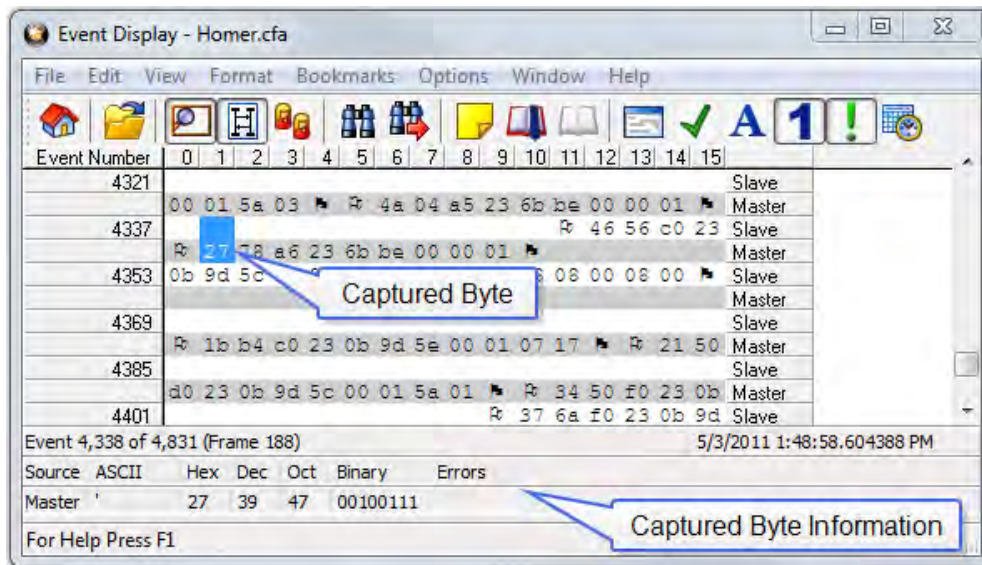




Figure 4.145 - Event Display

Click on an event to find out more about it. The three status lines at the bottom of the window are updated with information such as the time the event occurred (for data bytes, the time the byte was captured), the value of the byte in hex, decimal, octal, and binary, any errors associated with the byte, and more.















Events with errors are shown in red to make them easy to spot.

When capturing data live, the analyzer continually updates the Event Display as data is captured. Make sure the **Lock** icon  is displayed on the toolbar to prevent the display from updating (Clicking on the icon again will unlock the display). While locked, you can review your data, run searches, determine delta time intervals between bytes, and check CRCs. To resume updating the display, click the **Lock** icon again.

You can have more than one **Event Display** open at a time. Click the **Duplicate View** icon  to create a second, independent **Event Display** window. You can lock one copy of the **Event Display** and analyze your data, while the second **Event Display** updates as new data is captured.

Event Display is synchronized with the **Frame Display** and **Message Sequence Chart** dialogs. Selecting a byte in **Event Display** will also select the related frame in the **Frame Display** and the related message in the **Message Sequence Chart**.

4.7.2 The Event Display Toolbar

-  Home – Brings the Control window to the front.
-  Open a capture file
-  Start Capture - Begins data capture to disk.
-  Stop Capture - Closes a capture file and stops data capture to disk.
-  Save - Prompts user for a file name. If the user supplies a name, a .cfa file is saved.
-  Clear- Discards the temporary file and clears the display.
-  MSC Chart - Opens the Message Sequence Chart
-  Lock - In the Lock state, the window is locked so you can review a portion of data. Data capture continues in the background. Clicking on the Lock icon unlocks the window.
-  Unlock - In the Unlock state, the screen fills in the data captured since the screen lock and moves down to display incoming data again. Clicking on the Unlock icon locks the window.
-  Duplicate View - Creates a second Event Display window identical to the first.
-  Frame Display - (framed data only) Brings up a Frame Display, with the frame of the currently selected bytes highlighted.
-  Display Capture Notes - Brings up the Capture Notes window where you can view or add notes to the capture file.
-  Add/Modify Bookmark - Add a new or modify an existing bookmark.
-  Display All Bookmarks - Shows all bookmarks and lets you move between bookmarks.



Find - Search for errors, string patterns, special events and more.



Go To - Opens the Go To dialog, where you can specify which event number to go to.



CRC - Change the algorithm and seed value used to calculate CRCs. To calculate a CRC, select a byte range, and the CRC appears in the status lines at the bottom of the Event Display.



Mixed Sides - (Serial data only) By default, the analyzer shows data with the DTE side above the DCE side. This is called DTE over DCE format. DTE data has a white background and DCE data has a gray background. The analyzer can also display data in mixed side format. In this format, the analyzer does not separate DTE data from DCE data but shows all data on the same line as it comes in. DTE data is still shown with a white background and DCE data with a gray background so that you can distinguish between the two. The benefit of using this format is that more data fits onto one screen.



Character Only - The analyzer shows both the number (hex, binary, etc.) data and the character (ASCII, EBCDIC or BAUDOT) data on the same screen. If you do not wish to see the hex characters, click on the Character Only button. Click again to go back to both number and character mode.



Number Only - Controls whether the analyzer displays data in both character and number format, or just number format. Click once to show only numeric values, and again to show both character and numeric values.



All Events - Controls whether the analyzer shows all events in the window, or only data bytes. Events include control signal changes and framing information.




Timestamping Options – Brings up the timestamping options window which has options for customizing the display and capture of timestamps.

4.7.3 Opening Multiple Event Display Windows


Click the **Duplicate View** icon  from the **Event Display** toolbar to open a second **Event Display** window.


You can open as many **Event Display** windows as you like. Each **Event Display** is independent of the others and can show different data, use a different radix or character set, or be frozen or live.

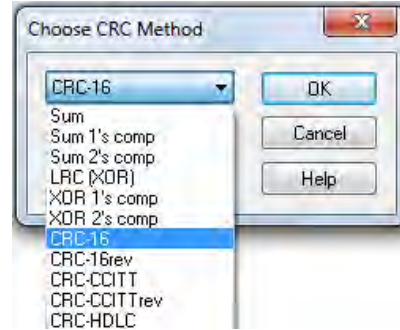
The **Event Display** windows are numbered in the title bar. If you have multiple **Event Displays** open, click on the **Event Display** icon  on the **Control** window toolbar to show a list of all the **Event Displays** currently open. Select a window from the list to bring it to the front.

4.7.4 Calculating CRCs or FCSs


The cyclic redundancy check (CRC) is a function on the **Event Display** window used to produce a checksum. The frame check sequence (FCS) are the extra checksum characters added to a frame to detect errors.

1. Open the **Event Display**  window.
2. Click and drag to select the data for which you want to generate a CRC.

3. Click on the **CRC** icon .
4. In the **CRC** dialog box, click on the down arrow to show the list of choices for CRC algorithms..
5. Enter a **Seed** value in hexadecimal if desired.
6. Click **OK** to generate the CRC. It appears in the byte information lines at the bottom of the Event Display window. Whenever you select a range of data, a CRC is calculated automatically.



4.7.5 Calculating Delta Times and Data Rates

1. Click on the **Event Display** icon  on the **Control** window to open the **Event Display** window.
2. Use the mouse to select the data you want to calculate a delta time and rate for.
3. The **Event Display** window displays the delta time and the data rate in the status lines at the bottom of the window.

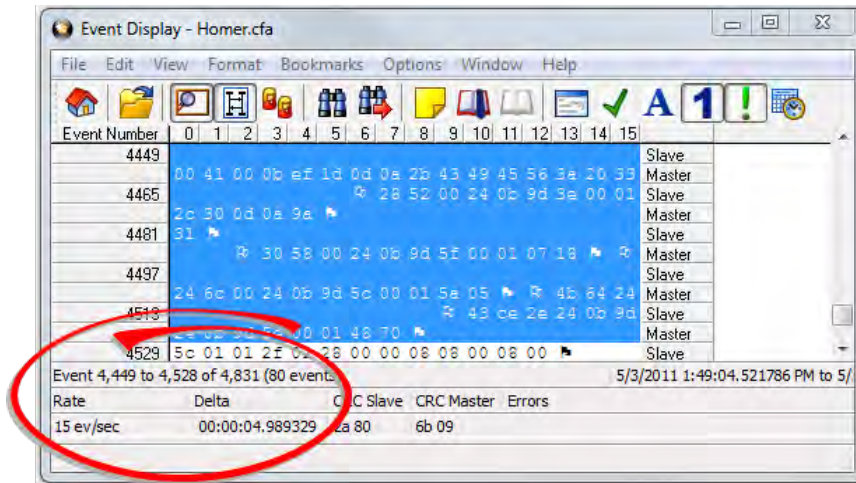





Figure 4.146 - Delta fields

4.7.6 Switching Between Live Update and Review Mode

The **Event Display** and **Frame Display** windows can update to display new data during live capture, or be frozen to allow data analysis. By default, the **Event Display** continually updates with new data, and the **Frame Display** is locked.

1. Make sure the **Lock** icon  is active so the display is locked and unable to scroll.
2. Click the **Unlock**  icon again to resume live update.

The analyzer continues to capture data in the background while the display is locked. Upon resuming live update, the display updates with the latest data.


You can have more than one **Event Display** or **Frame Display** window open at a time. Click the **Duplicate View** icon  to open additional Event or Frame Display windows. The lock/resume function is independent on each window. This means that you can have two **Event Display** windows open simultaneously, and one window can be locked while the other continues to update.

4.7.7 Data Formats and Symbols

4.7.7.1 Switching Between Viewing All Events and Viewing Data Events

By default, the analyzer on the Event Display dialog shows all **events**¹ that include:

- Data bytes
- Start-of-frame
- End-of-frame characters
- Data Captured Was Paused.

Click on the **Display All Events** icon  to remove the non-data events. Click again to display all events.

See [on page 267](#) for a list of all the special events shown in the analyzer and what they mean.

4.7.7.2 Switching Between Hex, Decimal, Octal or Binary

On the Event Display window the analyzer displays data in Hex by default. There are several ways to change the **radix**² used to display data.

Go to the **Format** menu and select the radix you want. A check mark next to the radix indicates which set is currently being used.

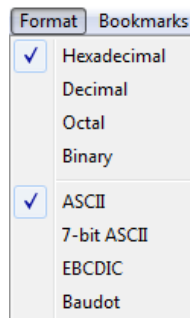


Figure 4.147 - Format Menu

¹An event is anything that happens on the circuit or which affects data capture. Data bytes, control signal changes, and long and short breaks are all events, as are I/O Settings changes and Data Capture Paused and Resumed.

²The base of a number system. Binary is base 2, octal is base 8, decimal is base 10 and hexadecimal is base 16.

1. Right-click on the data display header labels and choose a different radix.

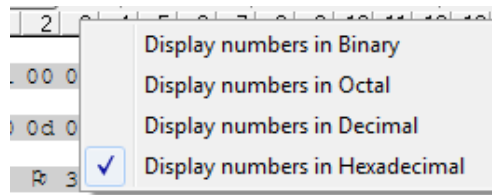


Figure 4.148 - Header labels, right click

2. Or right-click anywhere in the data display and select a different radix.

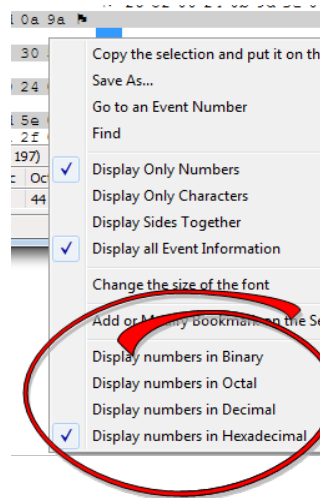





Figure 4.149 - Data display right click menu

If you want to see only the numerical values, click on the **Numbers Only** icon  on the **Event Display** toolbar.

4.7.7.3 Switching Between ASCII, EBCDIC, and Baudot


On the **Event Display** window, the analyzer displays data in ASCII by default when you click on the **Characters Only** icon . There are several ways to change the character set used to display data.

1. Go to the **Format** menu and select the character set you want. A check mark next to the character set indicates which set is currently being used.
2. With the data displayed in characters, right-click on the data panel header label to choose a different character set.


If you want to see only characters, click on the **Characters Only** icon  on the **Event Display** toolbar.

4.7.7.4 Selecting Mixed Channel/Sides

If you want to get more data on the **Event Display** window, you can switch to mixed sides mode. This mode puts all the data together on the same line. Data from one side (**Slave**) is shown on a white background and data from the other side (**Master**) is shown on a gray background.

1. Click once on the **Mixed Sides** icon  to put the display in mixed sides mode.
2. Click again to return to side over side mode.
3. You can right click in the center of the data display window to change between mixed and side over side modes by selecting **Display Sides Together**. A check mark is displayed. Click on **Display Sides Together** to remove the check mark and return to side-by-side display.
4. Right click in the sides panel on the right of the data display and select **Display Sides Together**. A check mark is displayed. Click on **Display Sides Together** to remove the check mark and return to side-by-side display.

4.7.7.5 List of all Event Symbols

By default, the **Event Display** shows all events, which includes control signal changes, start and end of frame characters and flow control changes. If you want to see only the data bytes, click on the All Events button . Click again to display all events.

Click on a symbol, and the analyzer displays the symbol name and sometimes additional information in the status lines at the bottom of the **Event Display** window. For example, clicking on a control signal change symbol displays which signal(s) changed.

In addition to data bytes, the events shown are (in alphabetical order):

Table 4.41 - Event Symbols





























| Symbol | Event |
|---|---|
|  | Abort |
|  | Broken Frame - The frame did not end when the analyzer expected it to. This occurs most often with protocols where the framing is indicated by a specific character, control signal change, or other data related event. |
|  | Buffer Overflow - Indicates a buffer overflow error. A buffer overflow always causes a broken frame. |
|  | Control Signal Change - One or more control signals changed state. Click on the symbol, and the analyzer displays which signal(s) changed at the bottom of the Event Display window. |
|  | Data Capture Paused - The Pause icon was clicked, pausing data capture. No data is recorded while capture is paused. |
|  | Data Capture Resumed - The Pause icon was clicked again, resuming data capture. |
|  | Dropped Frames - Some number of frames were lost. Click on the symbol, and the analyzer displays many frames were lost at the bottom of the Event Display window. |
|  | End of Frame - Marks the end of a frame. |
|  | Flow Control Active - An event occurred which caused flow control to become active (i.e. caused the analyzer to stop transmitting data) Events which activate flow control are signal changes or the receipt of an XON character. |

Table 4.41 - Event Symbols (continued)

| Symbol | Event |
|---|--|
|  | Flow Control Inactive - An event occurred which caused flow control to become inactive (i.e. caused the analyzer to transmit data). Events which deactivate flow control are signal changes or the receipt of an XOFF character. |
|  | Frame Recognizer Change - A lowest layer protocol was selected or removed here, causing the frame recognizer to be turned off or on. |
| \neq | I/O Settings Change - A change was made in the I/O Settings window which altered the baud, parity, or other circuit setting. |
|  | Long Break |
|  | Low Power - The battery in the ComProbe® is low. |
|  | Short Break |
|  | SPY Event (SPY Mode only) - SPY events are commands sent by the application being spied on to the UART. |
|  | Start of Frame - Marks the start of a frame. |
|  | Begin Sync Character Strip |
|  | End Sync Character Strip |
|  | Sync Dropped |
|  | Sync Found |
|  | Sync Hunt Entered |
|  | Sync Lost |
|  | Test Device Stopped Responding - The analyzer lost contact with the ComProbe for some reason, often because there is no power to the ComProbe. |
| $+$ | Test Device Began Responding - The analyzer regained contact with the ComProbe. |
|  | Timestamping Disabled - Timestamping was turned off. Events following this event are not timestamped. |
|  | Timestamping Enabled - Timestamping was turned on. Events following this event have timestamps. |
|  | Truncated Frame- A frame that is not the same size as indicated within its protocol. |
|  | Underrun Error |
|  | Unknown Event |

4.7.7.6 Font Size

The font size can be changed on several **Event Display** windows. Changing the font size on one window does not affect the font size on any other window.

To change the font size:

1. Click on **Event Display** menu **Options**, and select **Change the Font Size**.

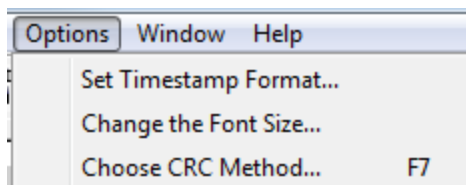


Figure 4.150 - Event Display Options menu

2. Choose a font size from the list.



Figure 4.151 - Event Display Font Size Selection

3. Click **OK**.

4.8 Data/Audio Extraction

You use Data/Audio Extraction to pull out data from various decoded *Bluetooth* protocols. Once you have extracted the data, you can save them into different file types, such as text files, graphic files, email files, .mp3 files, and more. Then you can examine the specific files information individually.

1. You access this dialog by selecting Extract Data/Audio from the View menu or by clicking on the icon from

the toolbar  .

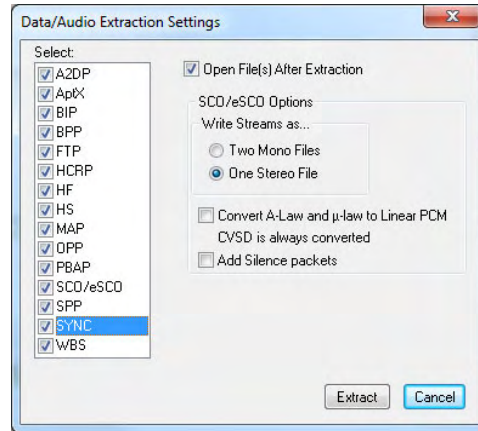


Figure 4.152 - Data/Audio Extraction Settings dialog

2. Choose a checkbox(es) on the left side of the dialog to identify from which profile(s) you want to extract data.

It's important to note that if there is no data for the profile(s) you select, no extracted file is created.

3. If you want the file(s) to open automatically after they are extracted, select the **Open File(s) After Extraction** checkbox.

Note: This does not work for SCO/eSCO.

4. Click on a radio button to write the streams as **Two Mono Files** or as **One Stereo File**.

Note: This option is for SCO/eSCO only.

5. Select the checkbox if you want to convert **A-Law and μ-law to Linear PCM**. CVSD are always converted to Linear PCM. It's probably a good idea to convert to Linear PCM since more media players accept this format.

Note: This option is for SCO/eSCO only.

6. Select the **Add Silence packets** to insert the silence packets (dummy packets) for the reserved empty slots into the extracted file. If this option is not selected, the audio packets are extracted without inserting the silence packets for the reserved empty slots.

Note: This option is for SCO/eSCO only.

7. Select **Extract**.

A **Save As** dialog appears.

The application will assign a file name and file type for each profile you select in Step 1 above. The file type varies depending on the original profile. A separate file for each profile will be created, but only for those profiles with available data.

8. Select a location for the file.

9. Click **Save**.

The **Data Extraction Status** and **Audio Extraction Status** dialogs appear. When the process is complete the dialogs display what files have been created and where they are located.

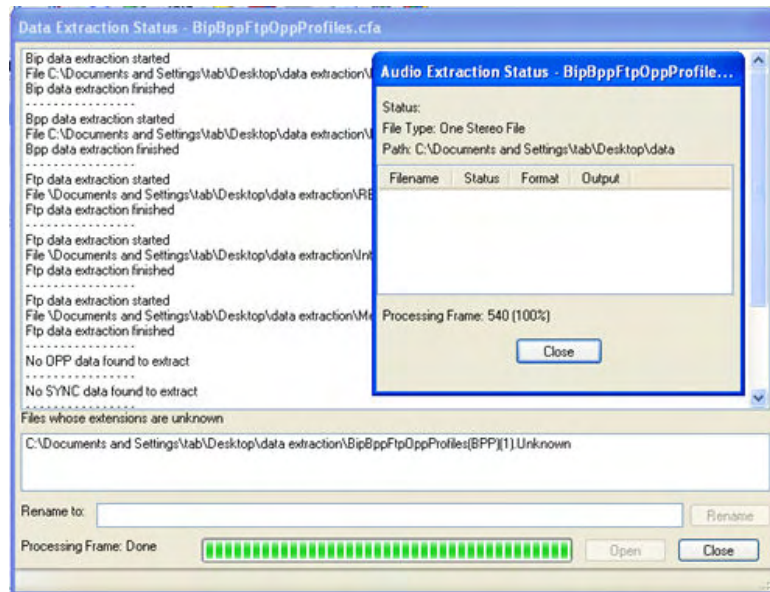
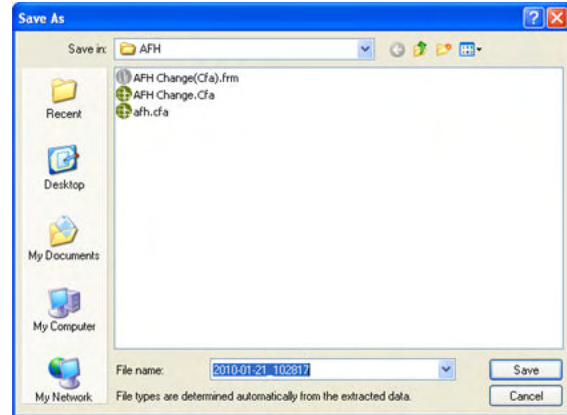


Figure 4.153 - Data and Audio Extraction Status

If you selected **Open Files(s) After Extraction**, the files open automatically.

10. If you did not select this option, you can open a file by simply double-clicking on the name.

Also, if a file type is unknown, you can select the file and it appears in the **Rename to:** text box.



Figure 4.154 - Rename To in the bottom section of Data Extraction Status

Then you can rename the file, adding a file type to attempt to open the file.

When you are finished, select **Close** to close the dialogs.

Chapter 5 Navigating and Searching the Data

The following sections describe how to navigate through the data and how to find specific data or packet conditions of interest to the user.

5.1 Find

Capturing and decoding data within the ComProbe analyzer produces a wealth of information for analysis. This mass of information by itself, however, is just that, a mass of information. There has to be ways to manage the information. ComProbe software provides a number of different methods for making the data more accessible. One of these methods is **Find**.

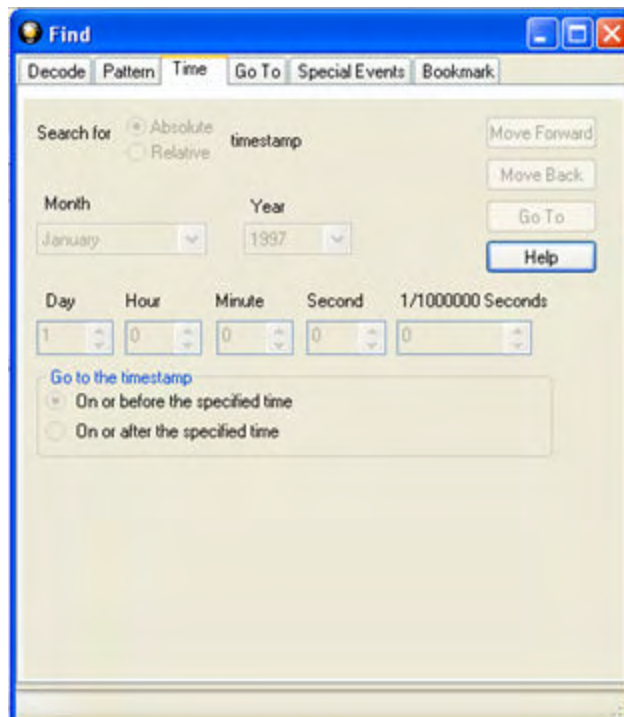





Figure 5.1 - Find Dialog

Find, as the name suggests, is a comprehensive search function that allows users to search for strings or patterns in the data or in the frame decode. You can search for errors, control signal changes, bookmarks, special events, time, and more. Once the information is located, you can easily move to every instance of the Find results.

5.1.1 Searching within Decodes

Searching within decodes lets you to do a string search on the data in the **Decode Pane** of the **Frame Display** window.

To access the search within decodes function:

1. Open a capture file to search.
2. Open the **Event Display**  or **Frame Display**  window.
3. Click on the **Find** icon  or choose **Find** from the **Edit** menu.
4. Click on the **Decode** tab of the **Find** dialog.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

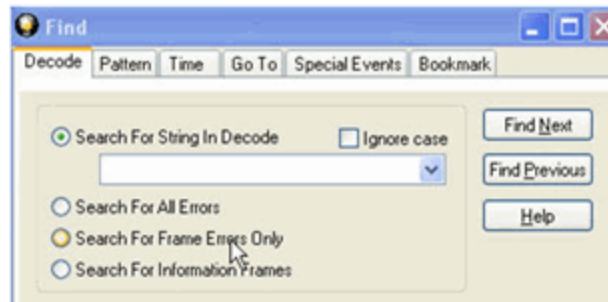


Figure 5.2 - Find Decode Tab Search for String

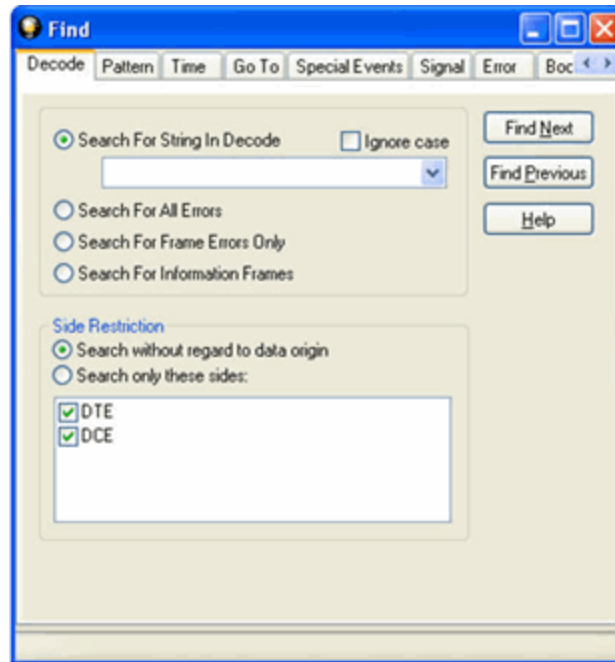


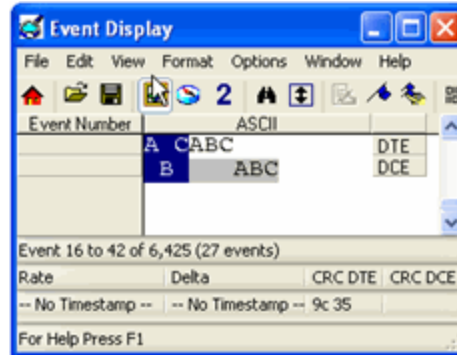
Figure 5.3 - Find Decode Tab Side Restriction

There are several options for error searching on the **Decoder** tab.

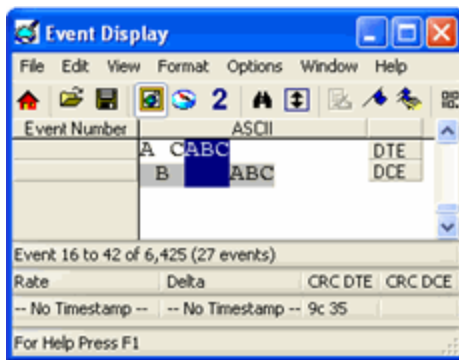
- **Search For String in Decoder** allows you to enter a string in the text box. You can use characters, hex or binary digits, wildcards or a combination of any of the formats when entering your string. Every time you type in a search string, the analyzer saves the search. The next time you open **Find**, the drop-down list will contain your search parameters.
- **Search for All Errors** finds frame errors as well as frames with byte-level errors (such as parity or CRC errors).
- **Search for Frame Errors Only** finds frame specific errors, such as frame check errors.
- **Search for Information Frame** only searches information frames.
 1. Enter the search string.
 2. Check **Ignore Case** to do a case-insensitive search.
 3. When you have specified the time interval you want to use, click on the **Find Next** or **Find Previous** buttons to start the search from the current event.

The result of the search is displayed in the **Decode** pane in **Frame Display**.

Side Restrictions - Side Restriction means that the analyzer looks for a pattern coming wholly from the DTE or DCE side. If you choose to search without regard for data origin, the analyzer looks for a pattern coming from one or both sides. For example, if you choose to search for the pattern ABC and you choose to search without regard for data origin, the analyzer finds all three instances of ABC shown here.



The first pattern, with the A and the C coming from the DTE device and the B coming from the DCE is a good example of how using a side restriction differs from searching without regard to data origin. While searching without regard for data origin finds all three patterns, searching using a side restriction never finds the first pattern, because it does not come wholly from one side or the other.



If you choose to search for the pattern ABC, and you restrict the search to just the DTE side, the analyzer finds the following pattern:

In this example, the analyzer finds only the second pattern (highlighted above) because we restricted the search to just the DTE side. The first pattern doesn't qualify because it is split between the DTE and DCE sides, and the third pattern, though whole, comes from just the DCE side.

If we choose both the DTE and the DCE sides in the above example, then the analyzer finds the second pattern followed by the third pattern, but not the first pattern. This is because each side has one instance in which the whole pattern can be found. The analyzer completely searches the DTE side first, followed by the DCE side.

Note: Side Restriction is available for pattern and error searching.



1. Select one of the two options.
2. Select **DTE**, **DCE**, or both.
3. When you made your selections, click on the **Find Next** or **Find Previous** buttons to start the search from the current event.


The result of the search is displayed in the **Decode** pane in **Frame Display**.

5.1.2 Searching by Pattern

Search by Pattern lets you perform a traditional string search. You can combine any of the formats when entering your string, and your search can include wildcards.

To access the search by pattern function:

1. Open a capture file to search.
2. Open the **Event Display**  or **Frame Display**  window.

3. Click on the **Find** icon  or choose **Find** from the **Edit** menu.
4. Click on the **Pattern** tab of the **Find** dialog.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

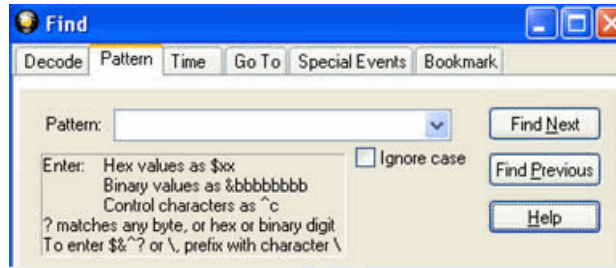


Figure 5.4 - Find Pattern Tab

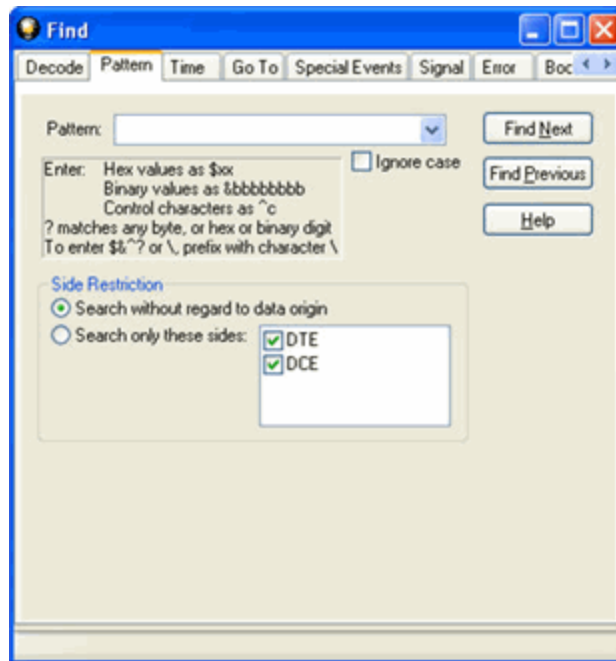


Figure 5.5 - Find Pattern Tab Side Restrictions

Pattern allows you to enter a string in the text box. You can use characters, hex or binary digits, control characters, wildcards or a combination of any of the formats when entering your string. Every time you type in a search string, the ComProbe analyzer saves the search. The next time you open **Find**, the drop-down list will contain your search parameters.

1. Enter the search pattern.
2. Check **Ignore Case** to do a case-insensitive search.

- When you have specified the pattern you want to use, click on the **Find Next** or **Find Previous** buttons to start the search from the current event.




The result of the search is displayed in the in Frame Display and Event Display.

Refer to Searching by Decode [on page 274](#) for information on **Side Restrictions**

5.1.3 Searching by Time

Searching with **Time** allows you search on timestamps on the data in **Frame Display** and **Event Display** window.

To access the search by time function:

- Open a capture file to search.
- Open the **Event Display**  or **Frame Display**  window.
- Click on the **Find** icon  or choose **Find** from the **Edit** menu.
- Click on the **Time** tab of the **Find** dialog.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

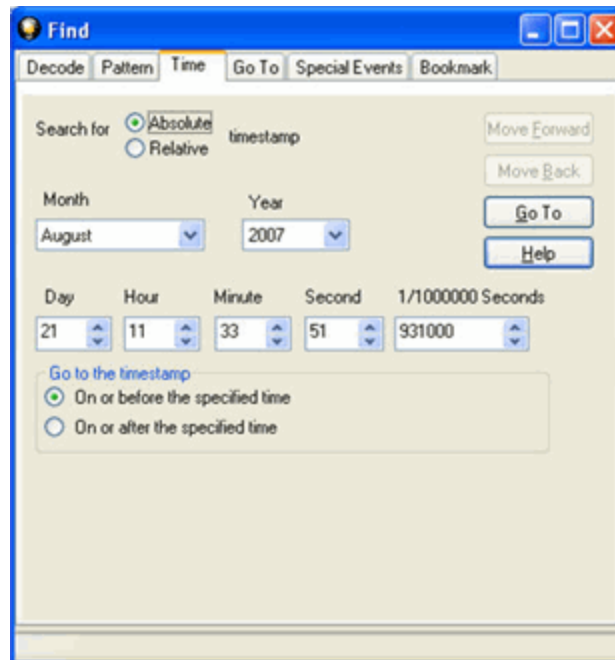


Figure 5.6 - Find by Time tab

The analyzer can search by time in several different ways.

Search for Absolute/Relative timestamp.

- **Absolute** - An absolute timestamp search means that the analyzer searches for an event at the exact date and time specified. If no event is found at that time, the analyzer goes to the nearest event either before or after the selected time, based on the "Go to the timestamp" selection.
- **Relative** - A relative search means that the analyzer begins searching from whatever event you are currently on, and search for the next event a specific amount of time away.

1. Select **Absolute** or **Relative**
2. Select the date and time using the drop-down lists for **Month, Year, Day, Hour, Minute, Second, 1/1000000**.

Note: Month and Year are not available if you select Relative.

3. When you have specified the time interval you want to use, click on the **Go To, Move Forward** or **Move Backward** buttons to start the search from the current event.

Note: When you select **Absolute** as **Search for**, **Go To** is available. When you select **Relative** as **Search for**, **Move Forward** or **Move Backward** is available.

Go to the timestamp: On or before/ On or after

The analyzer searches for an event that matches the time specified. If no event is found at the time specified, the analyzer goes to the nearest event either before or after the specified time. Choose whether to have the analyzer go to the nearest event before the specified time or after the specified time by clicking the appropriate radio button in the **Go to the timestamp** box.

If you are searching forward in the buffer, you usually want to choose the **On or After** option. If you choose the **On or Before** option, it may be that the analyzer finishes the search and not move from the current byte, if that byte happens to be the closest match.

When you select **Absolute** as **Search for**, the radio buttons are **On or before the specified time** or **On or after the specified time**. When you select **Relative** as **Search for**, the radio buttons are **On or before the specified time relative to the first selected item** or **On or after the specified time relative to the last selected item**.

1. Select **On or before the specified time** or **On or after the specified time**.
2. When you have specified the time interval you want to use, click on the **Go To, Move Forward** or **Move Backward** buttons to start the search from the current event.

When you select **Absolute** as **Search for**, **Go To** is available. When you select **Relative** as **Search for**, **Move Forward** or **Move Backward** is available.

There are a couple of other concepts to understand in respect to searching with timestamps.

- The analyzer skips some special events that do not have timestamps, such as frame markers. Data events that do not have timestamps because timestamping was turned off either before or during capture are also skipped.

- Timestamping can be turned on and off while data is being captured. As a result, the capture buffer may have some data with a timestamp, and some data without. When doing a search by timestamp, the analyzer ignores all data without a timestamp.
- The raw timestamp value is the number of 100-nanosecond intervals since the beginning of January 1, 1601. This is standard Windows time.

5.1.4 Using Go To

Searching with Go To allows you to go to a particular frame or event, or to move through the data X number of events or frames at a time. You can move either forward or backwards through the data.

To access the Go To function:

1. Open a capture file to search.
2. Open the **Event Display**  or **Frame Display**  window.
3. Click on the **Find** icon  or choose **Find** from the **Edit** menu.
4. Click on the **Go To** tab of the **Find** dialog.
5. The system displays the **Find** dialog with the **Go To** tab selected.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

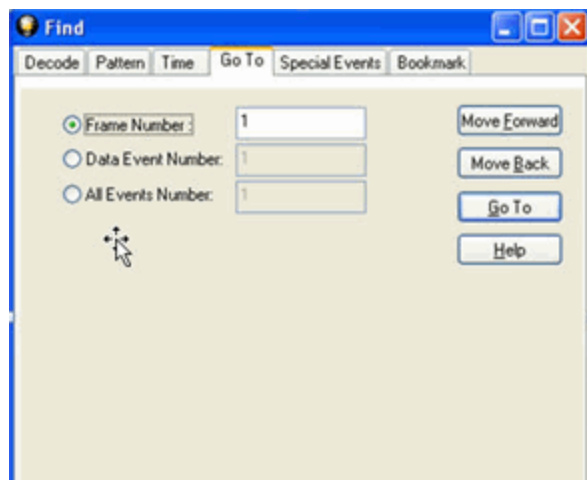


Figure 5.7 - Find Go To tab


To go to a particular frame :

1. Select the **Frame Number** radio button
2. Type the frame number in the box.
3. Click the **Go To** button.

4. To move forward or backward a set number of frames, type in the number of frames you want to move
5. Then click the **Move Forward** or **Move Back** button.

To go to a particular event :




1. Select the **Data Event Number** or **All Events Number** radio button.
2. Type the number of the event in the box.
3. Click the **Go To** button.
4. To move forward or backwards through the data, type in the number of events that you want to move each time.
5. Then click on the **Move Forward** or **Move Backward** button.
6. For example, to move forward 10 events, type the number 10 in the box, and then click on **Move Forward**. Each time you click on **Move Forward**, Frontline moves forward 10 events.

See [Event Numbering](#) for why the **Data Event Number** and **All Events Number** may be different. As a general rule, if you have the **Show All Events** icon  depressed on the **Event Display** window or **Frame Display Event** pane, choose **All Events Number**. If the **Show All Events** button is up, choose **Data Event Number**.

5.1.5 Searching for Special Events

Frontline inserts or marks events other than data bytes in the data stream. For example, the analyzer inserts start-of-frame and end-of-frame markers into framed data, marking where each frame begins and ends. If a hardware error occurs, the analyzer shows this using a special event marker. You can use Find to locate single or multiple special events.

To access the search for special events function:

1. Open a capture file to search.
2. Open the **Event Display**  or **Frame Display**  window.
3. Click on the **Find** icon  or choose **Find** from the **Edit** menu.
4. Click on the **Special Events** tab of the Find dialog.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

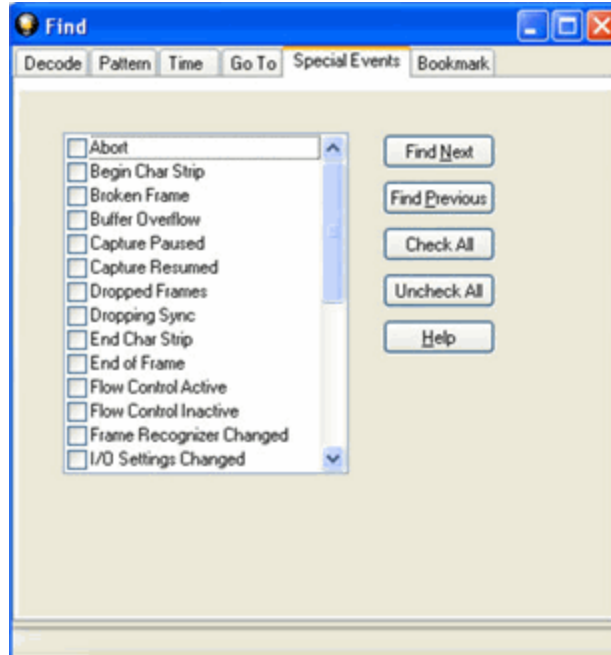


Figure 5.8 - Find Special Events tab

5. Check the event or events you want to look for in the list of special events. Use **Check All** or **Uncheck All** buttons to make your selections more efficient.
6. Click Find Next and Find Previous to move to the next instance of the event.

Not all special events are relevant to all types of data. For example, control signal changes are relevant only to serial data and not to Ethernet data.




For a list of all special events and their meanings, see [List of all Event Symbols on page 267](#).

5.1.6 Searching by Signal

Searching with Signal allows you to search for changes in control signal states for one or more control signals. You can also search for a specific state involving one or more control signals, with the option to ignore those control signals whose states you don't care about.

The analyzer takes the current selected byte as its initial condition when running searches that rely on finding events where control signals changed.

To access the search by time function:

1. Open a capture file to search.
2. Open the **Event Display**  or **Frame Display**  window.
3. Click on the **Find** icon  or choose **Find** from the **Edit** menu.
4. Click on the **Signal** tab of the **Find** dialog.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

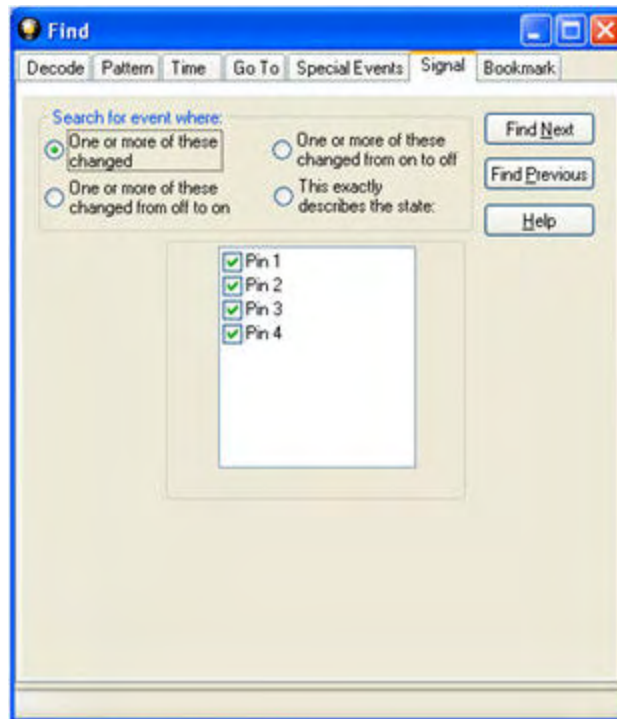


Figure 5.9 - Find Signal tab.

You will choose one qualifier—**Searching for event where**, then choose one or more control signals

Control Signals

The section with the check boxes allows you to specify which control signals the analyzer should pay attention to when doing the search. The analyzer pays attention to any control signal with a check mark.

- Click on a box to place a check mark next to a control signal
- Click again to uncheck the box
- By default, the analyzer searches all control signals, which means all boxes start out checked.

For example, if you are only interested in finding changes in **RTS** and **CTS**, you would check those two boxes and uncheck all the other boxes. This tells the analyzer to look only at the **RTS** and **CTS** lines when running the search. The other signals are ignored.

The control signals types include:

- USB - Pin 1
- USB - Pin 2
- USB - Pin 3
- USB - Pin 4

[Click here to learn more about the Breakout Box and Pins 1 - 4.](#)

Searching for event where:

- The first three options are all fairly similar, and are described together. These options are searching for an event where:
 - One or more control signals changed
 - One or more control signals changed from off to on
 - One or more control signals changed from on to off
- Searching for an event where one or more signals changed means that the analyzer looks at every control signal that you checked, and see if any one of those signals changed state at any time.
 - If you want to look at just one control signal:
 - Check the box for the signal.
 - Uncheck all the other boxes.
 - Choose to search for an event where one or more signals changed.
 - The analyzer notes the state of the selected signal at the point in the buffer where the cursor is, search the buffer, and stop when it finds an event where RTS changed state.
 - If the end of the buffer is reached before an event is found, the analyzer tells you that no matches were found.
- Searching for events where control signals changed state from off to on, or vice versa, is most useful if the signals are usually in one state, and you want to search for occasions where they changed state.

For example:

- If DTR is supposed to be on all the time but you suspect that DTR is being dropped
 - Tell the analyzer to look only at DTR by checking the DTR box and unchecking the others
 - Do a search for where one or more control signals changed from on to off.
 - The analyzer would search the DTR signal and stop at the first event where DTR dropped from on to off.
- Searching for an Exact State

To search for an exact state means that the analyzer finds events that match exactly the state of the control signals that you specify.




- First, choose to search for an event where your choices exactly describe the state.
- This changes the normal check boxes to a series of radio buttons labeled On, Off and Don't Care for each control signal.
- Choose which state you want each control signal to be in.
- Choose Don't Care to have the analyzer ignore the state of a control signal.
- When you click Find Next, the analyzer searches for an event that exactly matches the conditions selected, beginning from the currently selected event.

- If the end of the buffer is reached before a match is found, the analyzer asks you if you want to continue searching from the beginning.
- If you want to be sure to search the entire buffer, place your cursor on the first event in the buffer.
- Select one of the four radio buttons to choose the condition that must be met in the search
- Select one or more of the checkboxes for Pin 1, 2, 3, or 4.
- Click **Find Next** to locate the next occurrence of the search criteria or **Find Previous** to locate an earlier occurrence of the search criteria.

5.1.7 Searching for Data Errors

The analyzer can search for several types of data errors. Searching for data error allows you to choose which errors you want to search for and whether to search the DTE or DCE data or both. Bytes with errors are shown in red in the **Event Display** window, making it easy to find errors visually when looking through the data.

To access the search by time function:

1. Open a capture file to search.
2. Open the **Event Display**  or **Frame Display**  window.
3. Click on the **Find** icon  or choose **Find** from the **Edit** menu.
4. Click on the **Errors** tab of the **Find** dialog.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

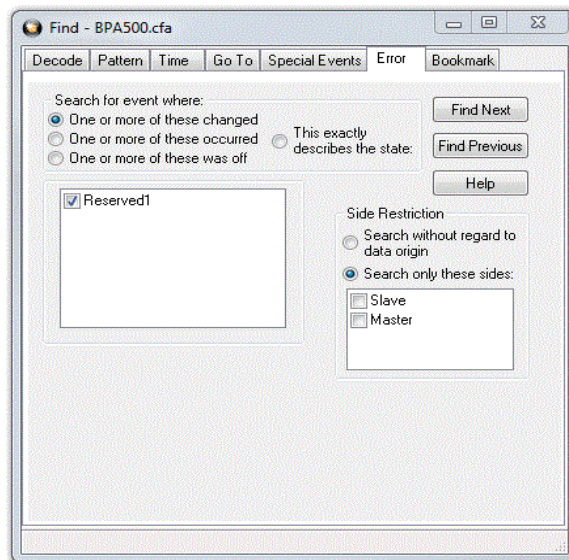


Figure 5.10 - Find Error tab.

Searching for event where

The first three options are all fairly similar, and are described together. These options are searching for an event where:

- one or more error conditions changed
- one or more error conditions occurred
- one or more error conditions were off (i.e. no errors occurred)

Selecting Which Errors to Search

The section with the check boxes allows you to choose which errors the analyzer should look for. Click on a box to check or un-check it.

If you want to search only for overrun errors

- check the box if shown
- un-check the other boxes.

To search for all types of errors

- check all boxes

The most common search is looking for a few scattered errors in otherwise clean data.

To do this type of search:

- choose to **Search for an event where** one or more error conditions occurred
- choose which errors to look for
- By default, the analyzer looks for all types of errors.

In contrast, searching for an event where one or more error conditions were off means that the analyzer looks for an event where the errors were not present.

For example, if you have data that is full of framing errors, and you know that somewhere in your 20 megabyte capture file the framing got straightened out, you could choose to search for an event where one or more error conditions were off, and choose to search only for framing. The analyzer searches the file, and finds the point at which framing errors stopped occurring.

Searching for an event where the error conditions changed means that the analyzer searches the data and stop at every point where the error condition changed from on to off, or off to on.

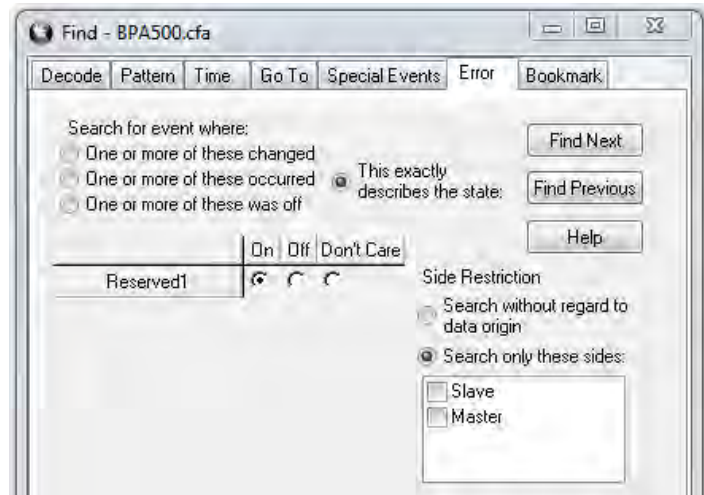
For example, if you have data where sometimes the framing is wrong and sometimes right, you would choose to search framing errors where the error condition changed. This first takes you to the point where the framing errors stopped occurring. When you click **Find Next**, the analyzer stops at the point when the errors began occurring again. Clicking **Find Previous** will search backwards from the current position.

The analyzer takes the current selected byte as its initial condition when running searches that rely on finding events where error conditions changed. The analyzer searches until it finds an event where error conditions changed or it reaches the end of the buffer, at which point the analyzer tells you that there are no more events found in the buffer. If you are searching for an exact match, the analyzer asks you if you want to continue searching from the beginning of the buffer.

Searching for Exact Error Conditions

To search for an exact state means that the analyzer finds events that exactly match the error conditions that you specify.

- Select the **This exactly describes the state** radio button.
- This changes the normal check boxes to a series of radio buttons labeled **On**, **Off** and **Don't Care** for each error.
 - **On** means that the error occurred
 - **Off** means that the error did not occur
 - **Don't Care** means that the analyzer ignores that error condition.
- Select the appropriate state for each type of error.



Example:

If you need to find an event where just an overrun error occurred, but not any other type of error, you would choose overrun error to be On, and set all other errors to Off. This causes the analyzer to look for an event where only an overrun error occurred.




If you want to look for events where overrun errors occurred, and other errors may have also occurred but it really doesn't matter if they did or not, choose overrun to be On, and set the others to Don't Care. The analyzer ignores any other type of error, and find events where overrun errors occurred.

To find the next error, click the Find Next button. To find an error that occurred earlier in the buffer to where you are, click the Find Previous button.

5.1.8 Find - Bookmarks

Searching with **Bookmarks** allows you search on specific [bookmarks](#) on the data in **Frame Display** and **Event Display** window. Bookmarks are notes/reminders of interest that you attach to the data so they can be accessed later.

To access the search for bookmarks

1. Open a capture file to search.
2. Open the **Event Display**  or **Frame Display**  window.
3. Click on the **Find** icon  or choose **Find** from the **Edit** menu.
4. Click on the **Bookmarks** tab of the **Find** dialog.

Note: The tabs displayed on the Find dialog depend on the product you are running and the content of the capture file you are viewing.

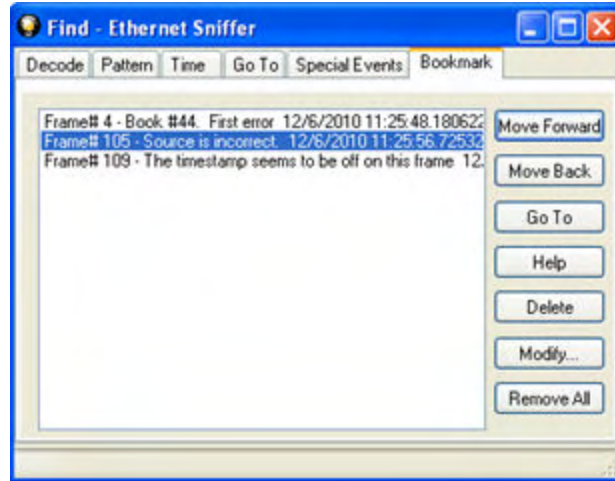



Figure 5.11 - Find Bookmark tab.

There are several ways to locate bookmarks.

- Select the bookmark you want to move to and click the **Go To** button.
- Simply double-click on the bookmark.
- Click the **Move Forward** and **Move Back** buttons to move through the frames to the bookmarks shown in the window. When the bookmark is found it is highlighted in the window.

There are three ways to modify bookmarks:

1. Click on **Delete** to remove the selected bookmark.
2. Click on **Modify...** to change the selected Bookmark name.
3. **Remove All** will delete all bookmarks in the window.

The **Find** window **Bookmark** tab will also appear when using functions other than **Find** such as when clicking on the Display All Bookmarks  icon.

5.1.9 Changing Where the Search Lands

When doing a search in the analyzer, the byte or bytes matching the search criteria are highlighted in the **Event Display**. The first selected byte appears on the third line of the display.

```
[CVEventDisplay]
SelectionOffset=2
```

To change the line on which the first selected byte appears:

1. Open fts.ini (located in the C:\User\Public\Public Documents\Frontline Test Equipment\)
2. Go to the [CVEventDisplay] section
3. Change the value for SelectionOffset.
4. If you want the selection to land on the top line of the display, change the SelectionOffset to 0 (zero).

5.1.10 Subtleties of Timestamp Searching

Timestamping can be turned on and off while data is being captured. As a result, the capture buffer may have some data with a timestamp, and some data without. When doing a search by timestamp, the analyzer ignores all data without a timestamp.

Note: The raw timestamp value is the number of 100-nanosecond intervals since the beginning of January 1, 1601. This is standard Windows time.

5.2 Bookmarks

Bookmarks are electronic sticky notes that you attach to frames of interest so they can be easily found later. In **Frame Display** bookmarked frames appear with a magenta triangle icon next to them.

| B... | Frame# | Command | Error Code | FID | MID | PID | Source | TID | UID | Fra... | Delta | Timestamp |
|------|--------|---------|------------|-----|-----|-----|--------|-----|-----|--------|---------------|--------------------|
| | 1 | | | | | | | | | 64 | | 12/6/2010 11:25... |
| | 2 | | | | | | | | | 168 | 00:00:00.0... | 12/6/2010 11:25... |
| ▶ E | 3 | | | | | | | | | 124 | 00:00:00.3... | 12/6/2010 11:25... |
| | 4 | | | | | | | | | 64 | 00:00:00.1... | 12/6/2010 11:25... |

Figure 5.12 - Bookmarked Frame (3) in the Frame Display

00 00 00 00 00 In the **Event Display** bookmarks appear as a dashed line around the start of frame
21 [P] 00 15 marker.

00 45 00 00 47

Bookmarks are easy to create and maintain, and are a very valuable tool for data analysis.


When you [create](#) or [modify](#) a bookmark, you have up to 84 characters to explain a problem, leave yourself a reminder, leave someone else a reminder, etc. Once you create a bookmark it will be saved with the rest of the data in the [.cfa file](#). When you open a .cfa file, the bookmarks are available to you.

Once you have created a bookmark, you can use the [Find](#) function or other navigation methods to [locate and move](#) among them.

5.2.1 Adding, Modifying or Deleting a Bookmark



You can add, modify, or delete a bookmarks from **Frame Display** and **Event Display**

Add:



1. Select the frame or event you want to bookmark.
2. There are three ways to access the **Add Bookmark** dialog.
 - a. Select **Add or Modify Bookmark** from the **Bookmarks** menu on the **Frame Display** and **Event Display**,
 - b. Select the **Add or Modify Bookmark**  icon on one of the toolbars, or
 - c. Right-click on the frame/event and choosing **Add Bookmark...**
3. In the dialog box, add a comment (up to 84 characters) in the text box to identify the bookmark.
4. Click **OK**.

Once you create a bookmark it will be saved with the rest of the data in the [.cfa file](#). When you open a .cfa file, the bookmarks are available to you.

Modify


1. Select the frame or event with the bookmark to be edited.
2. There are three ways to access the **Add/Modify Bookmark** dialog.
 - a. Select **Add or Modify Bookmark** from the **Bookmarks** menu on the **Frame Display** and **Event Display**
 - b. Select the **Add or Modify Bookmark**  icon on one of the toolbars, or
 - c. Right-click on the frame/event and choosing **Modify Bookmark...** on the selection.
3. Change the comment in the dialog box
4. Click **OK**. The edited bookmark will be saved as a part of the [.cfa file](#).
5. You can also select **Display All Bookmarks**  from the **Frame Display** and **Event Display** toolbar or the **Bookmarks** menu. the **Find** window will open on the **Bookmark** tab. Select the bookmark you want to modify and click the **Modify...** button. Change the comment in the dialog box, and click **OK**.

Delete

1. Select the frame or event with the bookmark to be deleted.
2. There are three ways to access the **Add/Modify Bookmark** dialog.
 - a. Select **Add or Modify Bookmark** from the **Bookmarks** menu on the **Frame Display** and **Event Display**,
 - b. Select the **Add or Modify Bookmark**  icon on one of the toolbars, or
 - c. Right-click on the frame/event and choosing **Modify Bookmark...** on the selection.
3. Click on the **Delete** button. The bookmark will be deleted.
4. You can also select **Display All Bookmarks**  from the **Frame Display** and **Event Display** toolbar or the **Bookmarks** menu. the **Find** window will open on the **Bookmark** tab. Select the bookmark you want to delete and click the **Delete** button.

5.2.2 Displaying All and Moving Between Bookmarks

There are three ways to move between bookmarks.

1. Press the F2 key to move to the next frame or event with a bookmark.
2. Select Go to Next Bookmark from the Bookmarks menu.
3. Click the Display All Bookmarks icon  . Select the bookmark you want to move to and click the Go To button, or simply double-click on the bookmark. Click the Move Forward and Move Back buttons to cycle through the bookmarks.

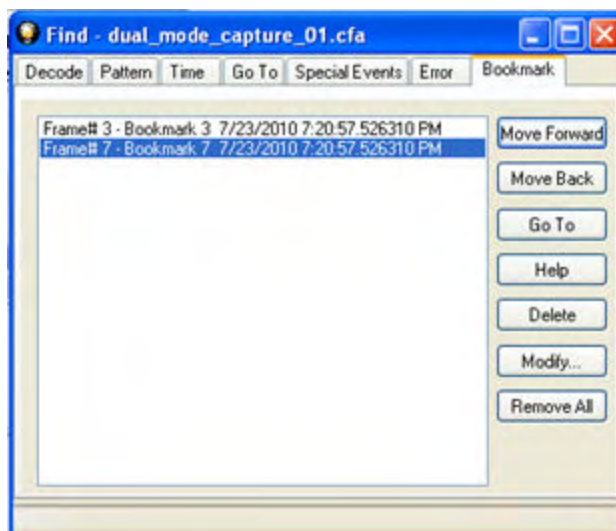


Figure 5.13 - Find Window Bookmark tab Used to Move Around With Bookmarks

To delete a bookmark, select it and click the **Delete** button.

To modify a bookmark, select it and click the **Modify** button.

Click **Remove All** to delete all the bookmarks.

Chapter 6 Saving and Importing Data

6.1 Saving Your Data





You can save all or part of the data that you have captured. You can also load a previously saved capture file, and save a portion of that file to another file. This feature is useful if someone else needs to see only a portion of the data in your capture file.

On the **Control** window toolbar you can set up to capture a single file. [Click here to see those settings.](#)

There are two ways to save portions or all of the data collected during a data capture. [Click here to see how to capture data to disk.](#)

6.1.1 Saving the Entire Capture File

This option is only available when you select **Single File** from the **Capture Mode** on **System Settings**. [Click here to learn more about selecting Save options from System Settings.](#)

1. If you are capturing data, click on the **Stop Capture**  icon to stop data capture. You cannot save data to file while it is being captured.
2. Open the **Event Display**  or **Frame Display**  window.
3. Click the **Save**  icon, or select **Save** from the **File** menu.

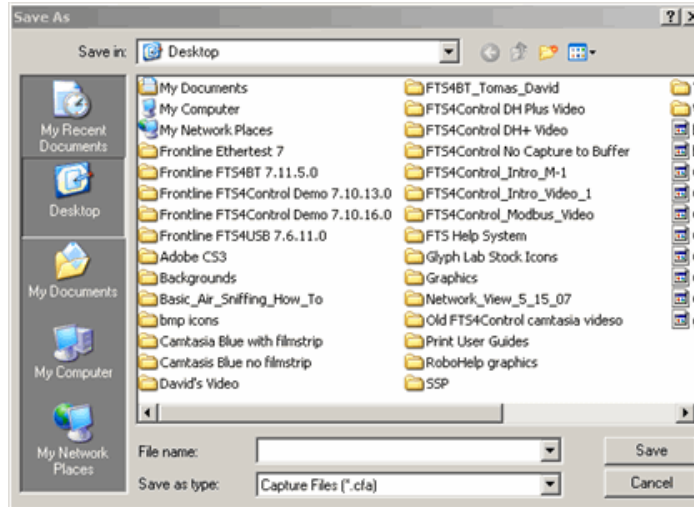



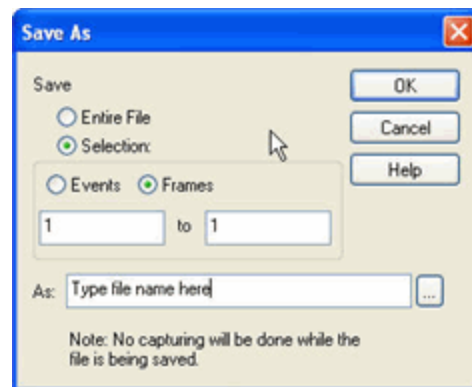


Figure 6.1 - Windows Save dialog




4. Type a file name in the **File name** box at the bottom of the screen.
5. Browse to select a specific directory. Otherwise your file is saved in the default capture file directory.
6. When you are finished, click **OK**.

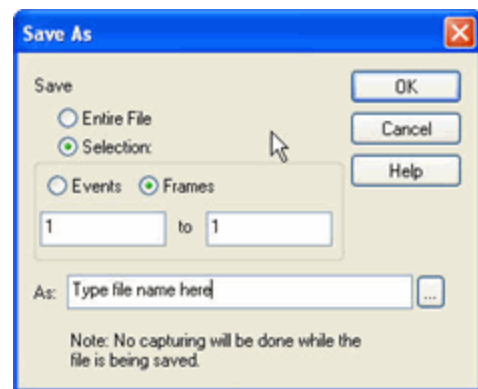
6.1.2 Saving the Entire Capture File with Save Selection

1. If you are capturing data, click on the **Stop** icon  to stop data capture. You cannot save data to file while it is being captured.
2. Open the **Event Display**  or **Frame Display**  window.
3. Right click in the data
4. Select **Save Selection** or **Save As** from the right click menu.
5. Click on the radio button labeled **Entire File**.
6. Choose to save **Events** or **Frames**. Choosing to save **Events** saves the entire contents of the capture file. Choosing to save **Frames** does not save all events in the capture file.
7. Type a file name in the **As** box at the bottom of the screen. Click the **Browse** icon to browse to a specific directory. Otherwise your file is saved in the default capture file directory.
8. When you are finished, click **OK**.



6.1.3 Saving a Portion of a Capture File




1. If you are capturing data, click on the **Stop** icon  to pause data capture. You cannot save data to a file while it is being captured.
2. Open the **Event Display**  or **Frame Display**  window, depending on whether you want to specify a range in bytes or in frames.
3. Select the portion of the data that you want to save. Click and drag to select data, or click on the first item, move to the last item and Shift+Click to select the entire range, or use the Shift key with the keyboard arrows or the navigation icons in the **Frame Display** toolbar. If the range you want to save is too large to select, note the numbers of the first and last item in the range.
4. Right click in the data
5. Select **Save Selection** or **Save As** from the right click menu
6. Click on the radio button labeled **Selection**. If you selected a range, make sure the starting and ending numbers are correct. To specify a range, type the numbers of the first and last items in the range in the boxes.
7. Select either **Events** or **Frames** to indicate whether the numbers are event or frame numbers.
8. Type a file name in the **As** box at the bottom of the screen. Click the **Browse** icon to browse to a specific directory. Otherwise your file is saved in the default capture file directory.
9. Click **OK** when you are finished.

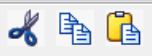




6.2 Adding Comments to a Capture File

The **Notes** feature allows you to add comments to a CFA file. These comments can be used for many purposes. For example, you can list the setup used to create the capture file, record why the file is useful to keep, or include notes to another person detailing which frames to look at and why. ([Bookmarks](#) are another useful way to record information about individual frames.)

To open the **Notes** window :

1. Click the **Show Notes** icon . This icon is present on the toolbars of the **Frame Display** , as well as the **Event Display** . **Notes** can be selected from the **Edit** menu on one of these windows.

2. Type your comments in the large edit box on the **Notes** window. The **Cut, Copy, Paste** features are supported from **Edit** menu and the toolbar  when text is selected. Undo and Redo features are all supported from **Edit** menu and the toolbar  at the current cursor location.
3. Click the thumbtack icon  to keep the **Notes** window on top of any other windows.
4. When you're done adding comments, close the window.
5. When you close the capture file, you are asked to confirm the changes to the capture file. See [Confirming Capture File \(CFA\) Changes](#) for more information.

6.3 Confirm Capture File (CFA) Changes

This dialog appears when you close a capture file after changing the [Notes](#), the protocol stack, or [bookmarks](#). The dialog lists information that was added or changed and allows you to select which information to save, and whether to save it to the current file or to a new one.


Changes made to the file appear in a list in the left pane. You can click on each item to see details in the right pane about what was changed for each item. You simply check the boxes next to the changes you want to keep. Once you decide what changes to keep, select one of the following:

- **Save To This File** – Saves the changes you have made to the current capture file.
- **Save As** – Saves the changes to a new file.
- **Cancel the Close Operation** – Closes the file and returns you back to the display. No changes are saved.
- **Discard Changes** – Closes the file without saving any of the changes made to the notes, bookmarks, or protocol stack.


6.4 Loading and Importing a Capture File

6.4.1 Loading a Capture File

From the Control Window:

1. Go to the **File** menu.
2. Choose a file from the recently used file list.
3. If the file is not in the **File** menu list, select **Open Capture File** from the **File** menu or simply click on the **Open** icon  on the toolbar.
4. Capture files have a .cfa extension. Browse if necessary to find your capture file.
5. Click on your file, and then click **Open**.

6.4.2 Importing Capture Files

1. From the **Control** window , go to the **File** menu and select Open Capture File or click on the Open icon on the toolbar.
2. Left of the **File name** text box, select from the drop-down list **Supported File Types** box to **All Importable File Types** or **All Supported File Types (*.cfa, *.log, *.txt, *.csv, *.cap)**. Select the file and click **Open**.

The analyzer automatically converts the file to the analyzer's format while keeping the original file in its original format. You can [save the file](#) in the analyzer's format, close the file without saving it in the analyzer's format, or have the analyzer automatically save the file in the analyzer's format (see the [System Settings](#) to set this option). All of these options keep your original file untouched.

When you first open the file, the analyzer brings up the [Protocol Stack](#) window and ask you what protocol decodes, if any, you want to use. You must choose a protocol decode at this point for the analyzer to decode the data in the file. If you open a file without using any decodes, and decide later that you want to apply a decode, choose [Reframe](#) from the File menu on the Control window.

At present, the analyzer supports the following file types:

- Frontline Serialtest* Async and Serialtest ComProbe® for DOS – requires the .byt for data and the .tim for timestamps (see note on importing [DOS timestamps](#)).
- Greenleaf ViewComm* 3.0 for DOS - requires the .byt for data and the .tim for timestamps (see note on importing [DOS timestamps](#)).
- Frontline Ethertest* for DOS – requires 3 files: filename.cap, filename.ca0 and filename.ca1.
- Sniffer Type 1 – supports files with the .enc extension. Does not support Sniffer files with a .cap extension.
- Snoop or Sun Snoop – files with a .cap extension based on RFC 1761. For file format, see <http://www.faqs.org/rfcs/rfc1761.html>.
- Shomiti Surveyor files in Snoop format – files with a .cap extension. For file format, contact [Technical Support](#).
- CATC Merlin - files with a .csv extension. Files must be exported with a specific format. See [File Format for Merlin Files](#) for information.
- CATC Chief - files with a .txt extension.

6.5 Printing

6.5.1 Printing from the Frame Display/HTML Export

The **Frame Display Print** dialog and the **Frame Display HTML Export** are very similar. This topic discusses both dialogs.

Frame Display Print

The **Frame Display Print** feature provides the user with the option to print the capture buffer or the current selection. The maximum file size, however, that can be exported is 1000 frames.

When **Print Preview** is selected, the output displays in a browser print preview window, where the user can select from the standard print options. The output file format is in html, and uses the Microsoft Web Browser Control print options for background colors and images.

Print Background Colors Using Internet Explorer

1. Open the Tools menu on the browser menu bar
2. Select “Internet Options...” menu entry.
3. Click Advanced tab.
4. Check “Print background colors and images” under the Printing section
5. Click the Apply button, then click OK

Configure the Print File Range in the Frame Display Print Dialog

Selecting more than one frame in the Frame Display window defaults the radio button in the Frame Display Print dialog to Selection and allows the user to choose the All radio button. When only one frame is selected, the All radio button in the Frame Display Print dialog is selected.

How to Print Frame Display Data

1. Select **Print** or **Print Preview** from the **File** menu on the **Frame Display** window to display the **Frame Display Print** dialog. Select **Print** if you just want to print your data to your default printer. Select **Print Preview** if you want access to printer options.
2. Choose to include the **Summary** pane (check the box) in the print output. The **Summary** pane appears at the beginning of the printed output in tabular format. If you select **All layers** in the **Detail Section**, the **Data Bytes** option becomes available.
3. In the **Detail Section**, choose to exclude—**No decode section**—the decode from the **Detail** pane in the **Frame Display**, or include **All Layers** or **Selected Layers Only**. If you choose to include selected layers, then select (click on and highlight) the layers from the list box.
4. Click on selected layers in the list to de-select, or click the **Reset Selected Layers** button to de-select all selected layers.

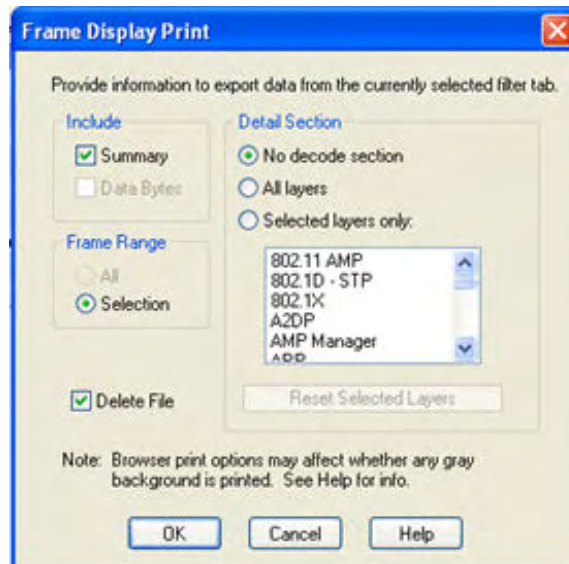


Figure 6.2 - Frame Display Print Dialog

5. Select the range of frames to include **All** or **Selection** in the **Frame Range** section of the **Frame Display Print** dialog.

Choosing **All** prints up to 1000 frames from the buffer.

Choosing **Selection** prints only the frames you select in the Frame Display window.

6. Selecting the **Delete File** deletes the temporary html file that was used during printing
7. Click the **OK** button.

Frame Display Print Preview

The **Frame Display Print Preview** feature provides the user with the option to export the capture buffer to an .html file. The maximum file size, however, that can be exported is 1000 frames.

If you chose **Print Preview**, the system displays your data in a browser print preview display with options for printing such as page orientation and paper size. You can also use your Printer Preferences dialog to make some of these selections. When printing your data, the analyzer creates an html file and prints the path to the file at the bottom of the page. This file can be opened in your browser, however, it may appear different than the printed version.

1. Select **Print Preview** from the **File** menu on the **Frame Display** window to display the **Frame Display Print Preview**.

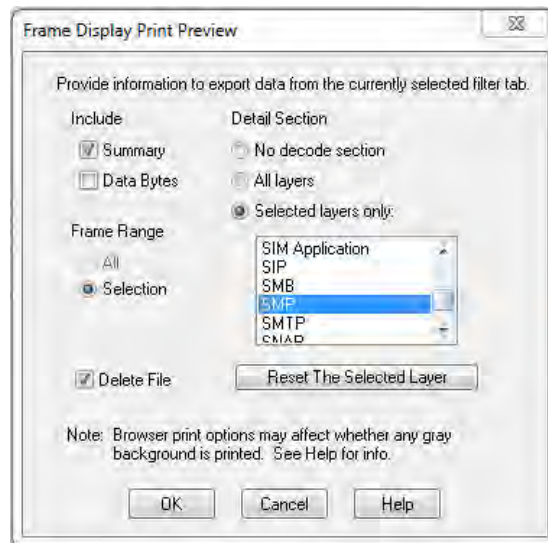


Figure 6.3 - Frame Display Print Preview Dialog

2. From this point the procedure is the same as steps 2 through 5 in "How to Print Frame Display Data" above.
3. Click the **OK** button, and after a brief wait a browser window will appear.

6.5.2 Printing from the Event Display

The Event Display Print feature provides the user with the option to print either the entire capture buffer or the current selection. When Print Preview is selected, the output displays in a browser print preview window where the user can select from the standard print options. The output file format is in html, and uses the Microsoft Web Browser Control print options for background colors and images (see below).

Print Background Colors Using Internet Explorer

1. Open the Tools menu on the browser menu bar
2. Select "Internet Options..." menu entry.
3. Click Advanced tab.
4. Check "Print background colors and images" under the Printing section
5. Click the Apply button, then click OK

The **Event Display Print** feature uses the current format of the **Event Display** as specified by the user.

See [About Event Display](#) for an explanation on formatting the **Event Display** prior to initiating the print feature.

Configure the Print File Range in the Event Display Print dialog

Selecting more than one event in the **Event Display** window defaults the radio button in the **Event Display Print** dialog to **Selection** and allows the user to choose the **All** radio button. When only one event is selected, the **All** radio button in the **Event Display Print** dialog is selected.

How to Print Event Display Data to a Browser

1. Select **Print** or **Print Preview** from the **File** menu on the **Event Display** window to display the **Event Display Print** dialog. Select **Print** if you just want to print your data to your default printer. Select **Print Preview** if you want preview the print in your browser.
2. Select the range of events to include from either **All** or **Selection** in the **Event Range** section. Choosing **All** prints all of the events in the capture file or buffer. Choosing **Selection** prints only the selected events in the Event Display window.

Note: In order to prevent a Print crash, you cannot select **All** if there are more than 100,000 events in the capture buffer.

Note: See "Configure the Print File Range in the Event Display Print Dialog" above for an explanation of these selections

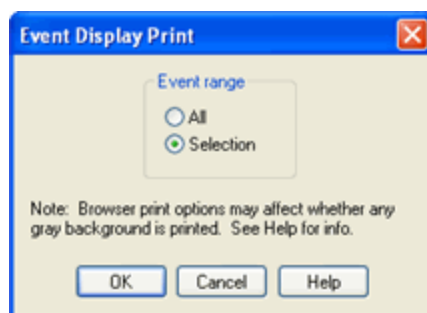


Figure 6.4 - Event Display Print Dialog

3. Click the OK button.

If you chose **Print Preview**, the system displays your data in a browser print preview display with options for printing such as page orientation and paper size. You can also use your Printer Preferences dialog to make some of these selections. When printing your data, the analyzer creates an html file and prints the path to the file at the bottom of the page. This file can be opened in your browser, however, it may appear different than the printed version.

6.6 Exporting

6.6.1 Frame Display Export

You can dump the contents of the **Summary** pane on the **Frame Display** into a Comma Separated File (.csv).

To access this feature:

1. Right click on the **Summary** pane or open the **Frame Display File** menu.
2. Select the **Export...** menu item.
3. Select a storage location and enter a **File name**.
4. Select **Save**.

6.6.2 Exporting a File with Event Display Export

With the **Event Display Export** dialog you can export the contents of the **Event Display** dialog as a text (.txt), CSV (.csv), HTML (.htm), or Binary File (.bin). You also have the option of exporting the entire capture buffer or just the current selection of the Event Display dialog.



Figure 6.5 - Event Display Export Example: .csv file.

How to Export Event Display Data to a File

1. Select **Export Events** from the **File** menu on the **Event Display** window to display the **Event Display Export** dialog.
2. Enter a file path and name, or click the browser button to display the Windows **Save As** dialog and navigate to the desired storage location.
3. Select a file type from the **Save as type:** drop-down List Menu on the Event Display Export dialog. Select from among the following file formats:

Text File (*.txt)

CSV File (*.csv)

HTML File (*.html)

Binary File (*.bin)

4. Select the range of events to include in the file from either **All** or **Selection** in the **Event Range** section of the **Event Display Export** dialog.
 - Selecting more than one event in the Event Display window defaults the radio button in the Event Display Export dialog to Selection and allows the user to choose the All radio button.
 - When only one event is selected (something must be selected), the All radio button in the Event Display Export dialog is selected by default.
5. Next you need to select the Side variable for serial communications.
 - is used to determine whether you want to export data from , or both.
 - Choose or Both to determine how you want to export the data.
5. Choose or Both to determine how you want to export the data.
6. Choose whether you want to display multiple events or single events per row.

Events Per Row: You can choose to display **Multiple Events Per Row**, but this method contains no timestamps. If you select **One Event Per Row**, you can display timestamps. multiple events or single events per row.

Note: The raw timestamp value is the number of 100-nanosecond intervals since the beginning of January 1, 1601. This is standard Windows time.

The timestamp data types displayed in columns for One Event Per Row.

Timestamp

Delta

Event Number

Byte Number

Frame Number

Type

Hex

Dec

Oct

Bin

Side

ASCII | 7-bit ASCII | EBCDIC | Baudot

RTS

CTS

DSR

DTR

CD

RI

UART Overrun

Parity Error

Framing Error

7. If you select .csv as the file type, choose whether you want to hide/display **Preambles** or **Column Headings** in the exported file
8. Click **Save**. The Event Display Export file is saved to the locations you specified in **File name**.

| 1 | Timestamp | Delta | Event Number | Byte Number | Frame Number | Type | Hex | Dec | Oct | Bin | ASCII |
|-----|-------------------------------|------------|--------------|-------------|--------------|--------|-----|-----|-----|----------|-------|
| 632 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 631 | 626 | | 3 Data | 0: | 0 | 0 | 0 | . |
| 633 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 632 | 627 | | 3 Data | 0: | 0 | 0 | 0 | . |
| 634 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 633 | 628 | | 3 Data | 0: | 0 | 0 | 0 | . |
| 635 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 634 | 629 | | 3 Data | 98: | 152 | 230 | 10011000 | . |
| 636 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 635 | 630 | | 3 Data | 70: | 112 | 160 | 11110000 | p |
| 637 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 636 | 631 | | 3 Data | 94: | 148 | 224 | 10010100 | . |
| 638 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 637 | 632 | | 3 Data | 22: | 34 | 42 | 100010 | " |
| 639 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 638 | 633 | | 3 Data | 21: | 33 | 41 | 100001 | ! |
| 640 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 639 | 634 | | 3 Data | 1c: | 28 | 34 | 111100 | . |
| 641 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 640 | 635 | | 3 Data | 80: | 128 | 200 | 10000000 | . |
| 642 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 641 | 636 | | 3 Data | 80: | 128 | 200 | 10000000 | . |
| 643 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 642 | 637 | | 3 Data | 80: | 128 | 200 | 10000000 | . |
| 644 | 11/30/2012 12:20:02.895166 PM | 0:00:00.00 | 643 | 638 | | 3 Data | 80: | 128 | 200 | 10000000 | . |

Figure 6.6 - Example: .csv Event Display Export, Excel spreadsheet

6.6.2.1 Export Filter Out

You can filter out data you don't want or need in your text file.

(This option is available only for serial data.) In the **Filter Out** box, choose which side to filter out: the DTE data, the DCE data or neither side (don't filter any data.) For example, if you choose the radio button for DTE data, the DTE data would be filtered out of your export file and the file would contain only the DCE data.

You can also filter out Special Events (which is everything that is not a data byte, such as control signal changes and Set I/O events), Non-printable characters or both. If you choose to filter out Special Events, your export file would contain only the data bytes. Filtering out the non-printable characters means that your export file would contain only special events and data bytes classified as printable. In ASCII, printable characters are those with hex values between \$20 and \$7e.

6.6.2.2 Exporting Baudot

When exporting Baudot, you need to be able to determine the state of the shift character. In a text export, the state of the shift bit can be determined by the data in the Character field. When letters is active, the character field shows letters and vice versa.

Chapter 7 General Information

7.1 System Settings and Program Options

7.1.1 System Settings

Open the **System Settings** window by choosing **System Settings** from the **Options** menu on the **Control** window. To enable a setting, click in the box next to the setting to place a checkmark in the box. To disable a setting, click in the box to remove the checkmark. When viewing a capture file, settings related to data capture are grayed out.

Single File

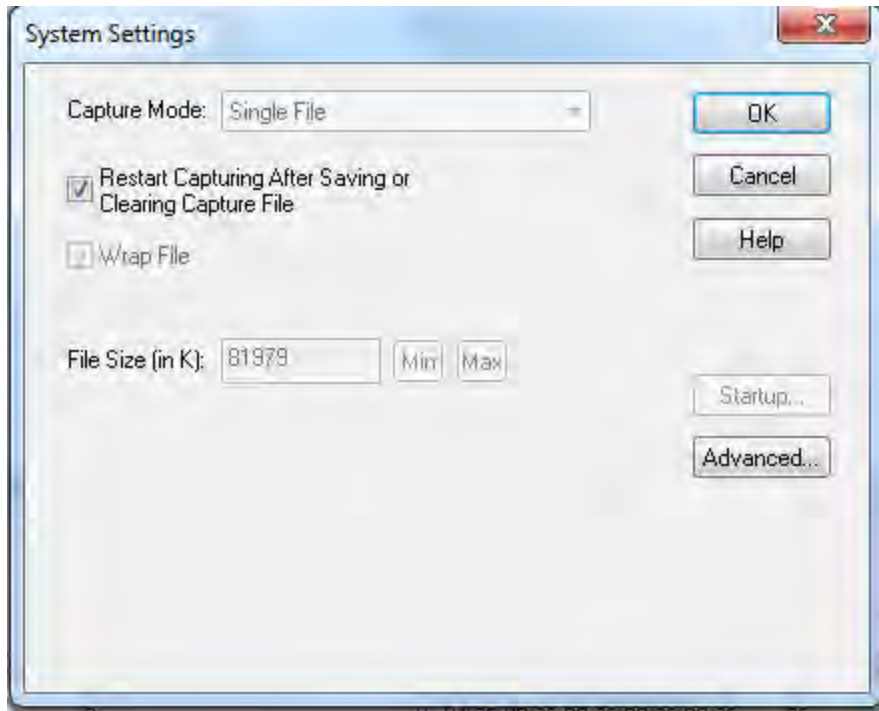


Figure 7.1 - System Settings Single File Mode

This option allows the analyzer to capture data to a file. Each time you capture the file you must provide a file name. The size of each file cannot larger than the number given in File Size (in K). The name of each file is the name you give it in the Name box followed by the date and time. The date and time are when the series was opened.

- **Restart Capturing After Saving or Clearing Capture File**

If the Automatically Restart feature is enabled, the analyzer restarts capture to the file immediately after the file is closed.

- **Wrap File**

When enabled, the analyzer wraps the file when it becomes full. The oldest events are moved out of the file to make room for new events. Any events moved out of the file are lost. When disabled, the analyzer stops capture when the file becomes full. Either reset the file or close your capture file to continue.

- **File Size:** The size of the file will depend of the available hard disk space.

1. Click the **Min** button to see/set the minimum acceptable value for the file size.
2. Click the **Max** button to see/set the maximum acceptable value for the file size.



You can accept these values, or you can enter a unique file size. But if you try to close the dialog after entering a value greater than the maximum or less than the minimum, you will see the following dialog.

- **Start up**

Opens the [Program Start up Options](#) window. **Start up** options let you choose whether to start data capture immediately on opening the analyzer.

- **Advanced**

Opens the [Advanced System Options](#) window. The Advanced Settings should only be changed on advice of technical support.

7.1.1.1 System Settings - Disabled/Enabled Options

Some of the **System Settings** options are disabled depending upon the status of the data capture session.


- As the default, all the options on the **System Settings** dialog are enabled.
- Once the user begins to capture data by selecting the Start Capture button, some of the options on the [System Settings](#) dialog are disabled until the user stops data capture and either saves or erases the captured data.
- The user can go into the [Startup options](#) and [Advanced system options](#) on the **System Settings** dialog and make changes to the settings at any time.

7.1.1.2 Advanced System Options

These parameters affect fundamental aspects of the software, and it is unlikely that you ever have to change them. If you do change them and need to return them to their original values, the default value is listed in parentheses to the right of the value box.

Most technical support problems are not related to these parameters, and as changing them could have serious consequences for the performance of the analyzer, we strongly recommend contacting technical support before changing any of these parameters.

To access the Advanced System Options:

1. Go to the Control  window.
2. Choose **System Settings** from the **Options** menu.
3. On the **System Settings** window, click the **Advanced** button.

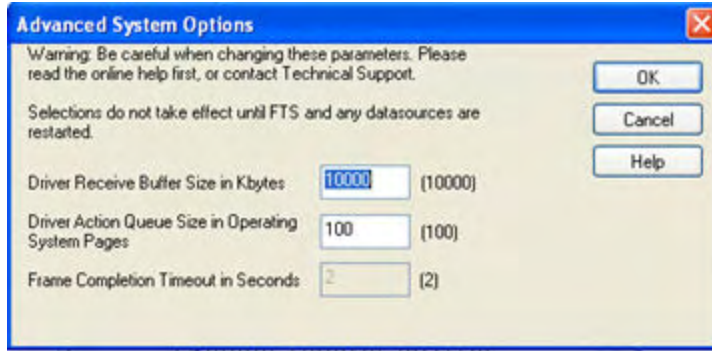


Figure 7.2 - Advanced System Options dialog

- **Driver Receive Buffer Size in Kbytes** - This is the size of the buffer used by the driver to store incoming data. This value is expressed in Kbytes.
- **Driver Action Queue Size In Operating System Pages** - This is the size of the buffer used by the driver to store data to be transmitted. This value is expressed in operating system pages.
- **Frame Completion Timeout in Seconds** - This is the number of seconds that the analyzer waits to receive data on a side while in the midst of receiving a frame on that side.


If no data comes in on that side for longer than the specified number of seconds, an "aborted frame" event is added to the Event Display and the analyzer resumes decoding incoming data. This can occur when capturing interwoven data (DTE and DCE) and one side stops transmitting in the middle of a frame.

The range for this value is from 0 to 999,999 seconds. Setting it to zero disables the timeout feature.

Note: This option is currently disabled.

7.1.1.3 Selecting Start Up Options

To open this window:

1. Choose **System Settings** from the **Options** menu on the Control  window.
2. On the System Settings window, click the **Start Up** button.
3. Choose one of the options to determine if the analyzer starts data capture immediately on starting up or not.

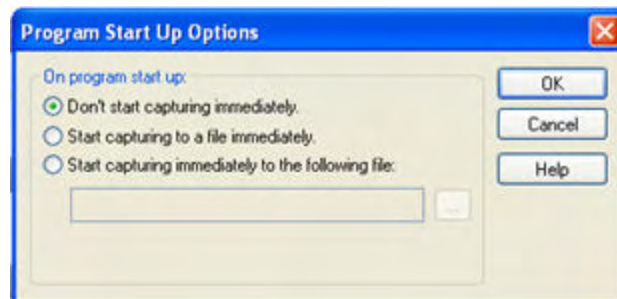




Figure 7.3 - Start Up Options dialog

- **Don't start capturing immediately** - This is the default setting. The analyzer begins monitoring data but does not begin capturing data until clicking the **Start Capture**  icon on the **Control, Event Display** or **Frame Display** windows.
- **Start capturing to a file immediately** - When the analyzer starts up, it immediately opens a capture file and begins data capture to it. This is the equivalent of clicking the **Start Capture**  icon. The file is given a name based on the settings for capturing to a file or series of files in the **System Settings** window.
- **Start capturing immediately to the following file:** - Enter a file name in the box below this option. When the analyzer starts up, it immediately begins data capture to that file. If the file already exists, the data in it is overwritten.

7.1.2 Changing Default File Locations

The analyzer saves user files in specific locations by default. Capture files are placed in the My Capture Files directory and configurations are put in My Configurations. These locations are set at installation.

Follow the steps below to change the default locations.

1. Choose **Directories** from the **Options** menu on the **Control** window to open the **File Locations** window.

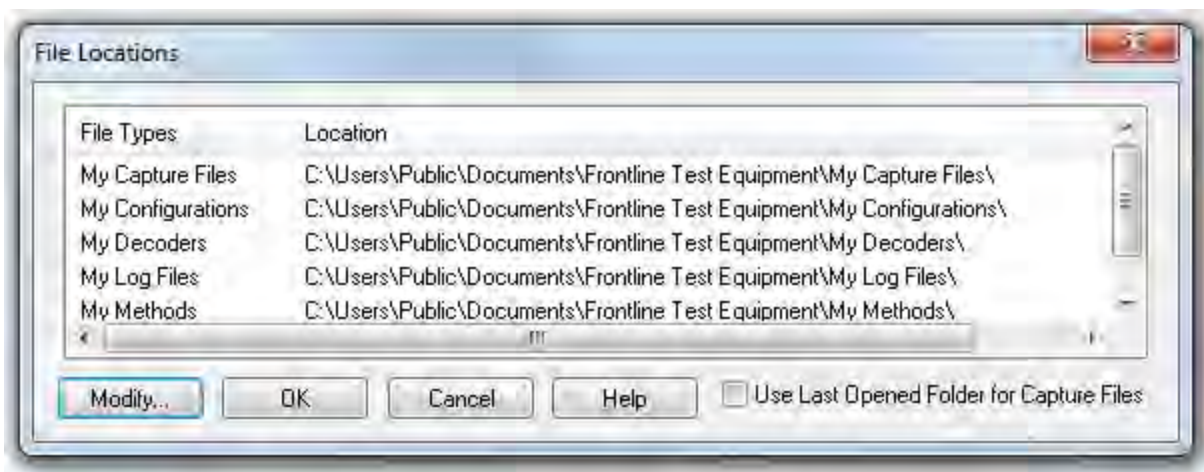


Figure 7.4 - File Locations dialog

2. Select the default location you wish to change.
3. Click **Modify**.
4. Browse to a new location.

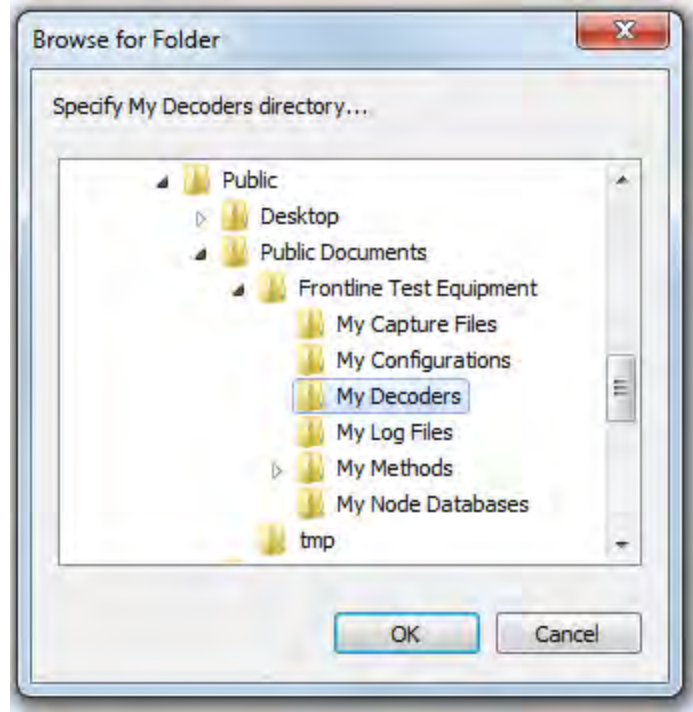


Figure 7.5 - File Locations Browse dialog

5. Click **OK**.
6. Click **OK** when finished.

If a user sets the My Decoders directory such that it is up-directory from an installation path, multiple instances of a personality entry may be detected, which causes a failure when trying to launch Frontline. For example, if an Frontline product is installed at C:\Users\Public\Public Documents\Frontline Test Equipment\My Decoders\ then "My Decoders" cannot be set to any of the following:

- C:\ My Decoders\
- C:\Users\ My Decoders\
- C:\Users\Public\My Decoders\
- C:\Users\Public\Public Documents\My Decoders\
- or to any directory that already exists in the path C:\Users\Public\Public Documents\Frontline Test Equipment\My Decoders\

Default Capture File Folder Checkbox

If the **Use Last Opened Folder for Capture Files** checkbox is checked, then the system automatically changes the default location for saving capture files each time you open a file from or save a file to a new location. For example, let's say the default location for saving capture files is Drive A > Folder A. Now you select the **Use Last Opened Folder for Capture Files** checkbox. The next time, however, you open a capture file from a different location, Folder B > Removable Flash Drive for example. Now when you save the capture file, it will be saved to Folder B > Removable Flash Drive. Also, all subsequent files will be saved to that location. This remains true until you open a file from or save a file to a different location.

There is one caveat to this scenario, however. Let's say you have selected **Use Last Opened Folder for Capture Files** and opened a file from a location other than the default directory. All subsequent capture files will be saved to that location. Suppose, however, the next time you want to save a capture file, the new file location is not available because the directory structure has changed: a folder has been moved, a drive has been reassigned, a flash drive has been disconnected, etc. In the case of a "lost" directory structure, subsequent capture files will be saved to the default location. **ComProbe software will always try to save a file to the folder where the last file was opened from or saved to, if Use Last Opened Folder for Capture Files is checked.** If, however, the location is not accessible, files are saved to the default directory that is set at installation.

If the checkbox is unchecked, then the system always defaults to the directory listed in the File Locations dialog.

7.1.3 Side Names

The **Side Names** dialog is used to change the names of objects and events that appear in various displays. **The Side Names** dialog will change depending on the sniffing technology in use at the time the software was loaded.

Changes to the Names are used throughout the program.



Figure 7.6 - Example: Side Names Where "Slave" and "Master" are current

1. To open the Side Names dialog, choose **Side Names...** from the **Options** menu on the **Control** window.
2. To change a name, click on the name given in the **Current Names** column, and then click again to modify the name (a slow double-click).
3. Select **OK** to initiate the changes. The changes that have been made will not fully take effect for any views already open. Closing and reopening the views will cause the name change to take effect.
4. To restore the default values, click the **Set Defaults** button.


7.1.4 Timestamping

Timestamping is the process of precise recording in time of packet arrival. Timestamps is an optional parameter in the Frame Display and Event Display that can assist in troubleshooting a network link.

7.1.4.1 Timestamping Options

The Timestamping Options window allows you to enable or disable timestamping, and change the resolution of the timestamps for both capture and display purposes.

To open this window:

Choose **Set Timestamp Format...** from the **Options** menu on the Frame Display and Event Display window or click on the **Timestamping Option**  icon in the **Event Display** toolbar. The Timestamping Options window will open.

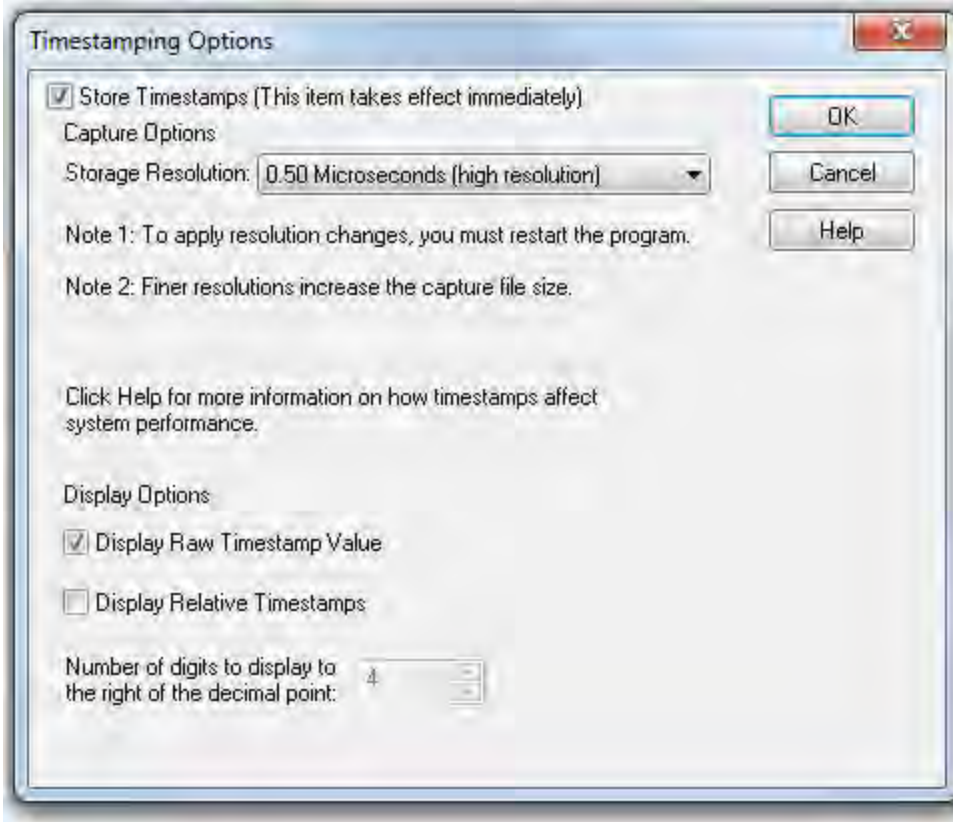


Figure 7.7 - Timestamping Options dialog

Enabling/Disabling Timestamp

To enable timestamping click to make a check appear in the check box **Store Timestamps (This time takes effect immediately)**. Removing the check will disable timestamping.

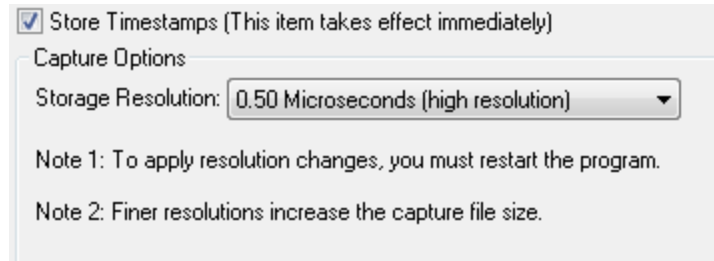
Changing the Timestamp Resolution

This option affects the resolution of the timestamp stored in the capture file. The default timestamp is 10 milliseconds. This value is determined by the operating system and is the smallest "normal" resolutions possible.

Note: The raw timestamp value is the number of 100-nanosecond intervals since the beginning of January 1, 1601. This is standard Windows time.

It is also possible to use "high resolution" timestamping. High resolution timestamp values are marked by an asterisk as high resolution in the drop down list. To change timestamping resolutions:

1. Go to the **Capture Options** section of the window.
2. Change the resolution listed in the **Storage Resolution** box.



Note: If you change the resolution, you need to exit the analyzer and restart in order for the change to take effect.

Performance Issues with High Resolution Timestamp



There are two things to be aware of when using high resolution timestamps. The first is that high resolution timestamps take up more space in the capture file because more bits are required to store the timestamp. Also, more timestamps need to be stored than at normal resolutions. The second issue is that using high resolution timestamping may affect performance on slower machines

For example, if 10 bytes of data are captured in 10 milliseconds at a rate of 1 byte per millisecond, and the timestamp resolution is 10 milliseconds, then only one timestamp needs to be stored for the 10 bytes of data. If the resolution is 1 millisecond, then 10 timestamps need to be stored, one for each byte of data. If you have two capture files, both of the same size, but one was captured using normal resolution timestamping and the other using high resolution, the normal resolution file has more data events in it, because less room is used to store timestamps.

You can increase the size of your capture file in the [System Settings](#).

Switching Between Relative and Absolute Time

With Timestamping you can choose to employ Relative Time or Absolute time.

1. Choose **System Settings** from the **Options** menu on the **Control** window, and click the **Timestamping Options** button, or click the **Timestamping Options** icon  from the **Event Display**  window.
2. Go to the **Display Options** section at the bottom of the window and find the **Display Relative Timestamps** checkbox.
3. Check the box to switch the display to relative timestamps. Remove the check to return to absolute timestamps.




Note: The options in this section affect only how the timestamps are displayed on the screen, not how the timestamps are recorded in the capture file.

- **Display Raw Timestamp Value** shows the timestamp as the total time in hundred nanoseconds from a specific point in time.

- **Display Relative Timestamps** shows the timestamp as the amount of time that has passed since the first byte was captured. It works just like a stop watch in that the timestamp for the first byte is 0:00:00.0000 and all subsequent timestamps increment from there. The timestamp is recorded as the actual time, so you can flip back and forth between relative and actual time as needed.
- Selecting both values displays the total time in nanoseconds from the start of the capture as opposed to a specific point in time.
- Selecting neither value displays the actual chronological time.

When you select **Display Relative Timestamp** you can set the number of digits to display using the up or down arrows on the numeric list.

Displaying Fractions of a Second

1. Choose **System Settings** from the **Options** menu on the **Control**  window, and click the **Timestamping Options** button, or click the **Timestamping Options** icon  from the **Event Display**  window.
2. Go to the **Display Options** section at the bottom of the window, and find the **Number of Digits to Display** box.
3. Click on the arrows to change the number. You can display between 0 and 6 digits to the right of the decimal point.

7.2 Technical Information

7.2.1 Performance Notes

As a software-based product, the speed of your computer's processor affects the analyzer's performance. Buffer overflow errors are an indicator that the analyzer is unable to keep up with the data. The information below describes what happens to the data as it arrives, what the error means, and how various aspects of the analyzer affect performance. Also included are suggestions on how to improve performance.

The analyzer's driver takes data from the driver and counts each byte as they are put into the driver's buffer. The analyzer's driver tells the user interface that data is ready to be processed. The analyzer takes the data from the driver's buffer and puts the data into the capture buffer.

Driver Buffer Overflows occur when the user interface does not retrieve frames from the driver quickly enough. Buffer overflows are indicated in the **Event Display** window by a plus sign within a circle. Clicking on the buffer overflow symbol displays how many frames have been lost.

There are several things that you can do to try and solve this problem.

- Use capture filters to filter out data you don't need to see. Capture filters reduce the amount of data processed by the analyzer. (Ethernet Only)
- Close all other programs that are doing work while the analyzer is running. Refrain from doing searches in the **Event Display** window or other processor intensive activities while the analyzer is capturing data.
- Timestamping takes up processor time, primarily not in timestamping the data, but in writing the timestamp to the file. Try turning off timestamping from the [Timestamping Options](#) window.

- For **Driver Buffer Overflows**, change the size of the driver buffer. This value is changed from the **Advanced System Settings**. Go to the **Control** window and choose **System Settings** from the **Options** menu. Click on the **Advanced** button. Find the value **Driver Receive Buffer Size in Operating System Pages**. Take the number listed there and double it.
- The analyzer's number one priority is capturing data; updating windows is secondary. However, updating windows still takes a certain amount of processor time, and may cause the analyzer to lose data while the window is being updated. Some windows require more processing time than others because the information being displayed in them is constantly changing. Refrain from displaying data live in the **Event Display** and **Frame Display** windows. The analyzer can capture data with no windows other than the **Control** window open.
- If you are still experiencing buffer overflows after trying all of the above options, then you need to use a faster PC.

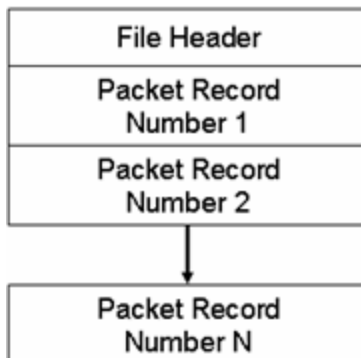
7.2.2 BTSnoop File Format

Overview

The BTSnoop file format is suitable for storing Bluetooth® HCI traffic. It closely resembles the snoop format, as documented in RFC 1761.

File Format

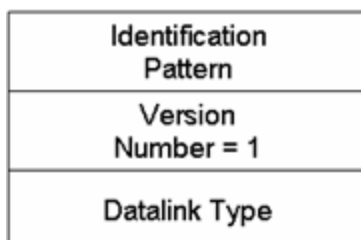
The snoop packet capture file is an array of octets structured as follows:



The File Header is a fixed-length field containing general information about the packet file and the format of the packet records it contains. One or more variable-length Packet Record fields follow the File Header field. Each Packet Record field holds the data of one captured packet.

File Header

The structure of the File Header is as follows:



Identification Pattern:

A 64-bit (8 octet) pattern used to identify the file as a snoop packet capture file. The Identification Pattern consists of the 8 hexadecimal octets:

62 74 73 6E 6F 6F 70 00

This is the ASCII string "btsnoop" followed by one null octets.

Version Number:

A 32-bit (4 octet) unsigned integer value representing the version of the packet capture file being used. This document describes version number 1.

Datalink Type:

A 32-bit (4 octet) field identifying the type of datalink header used in the packet records that follow. The datalink type codes are listed in the table below. Values 0 - 1000 are reserved, to maximize compatibility with the RFC1761 snoop version 2 format.

Table 7.1 - Datalink Codes

| Datalink Type | Code |
|--------------------------|-------------------|
| Reserved | 0 - 1000 |
| Un-encapsulated HCI (H1) | 1001 |
| HCI UART (H4) | 1002 |
| HCI BSCP | 1003 |
| HCI Serial (H5) | 1004 |
| Unassigned | 1005 - 4294967295 |

Packet Record Format

Each packet record holds a partial or complete copy of one packet as well as some descriptive information about that packet. The packet may be truncated in order to limit the amount of data to be stored in the packet file.

Each packet record holds 24 octets of descriptive information about the packet, followed by the packet data, which is variable-length, and an optional pad field. The descriptive information is structured as six 32-bit (4-octet) integer values.

The structure of the packet record is as follows:

| |
|------------------------|
| Original Length |
| Included Length |
| Packet Flags |
| Cumulative Drops |
| Timestamp Microseconds |
| Packet Data |

Original Length

A 32-bit unsigned integer representing the length in octets of the captured packet as received via a network.

Included Length

A 32-bit unsigned integer representing the length of the Packet Data field. This is the number of octets of the captured packet that are included in this packet record. If the received packet was truncated, the Included Length field is less than the Original Length field.

Packet Flags

Flags specific to this packet. Currently the following flags are defined:

Table 7.2 - Packet Flag Description

| Bit No. | Definition |
|---------|--|
| 0 | Direction flag 0 = Sent, 1 = Received |
| 1 | Command flag 0 = Data, 1 = Command/Event |
| 2 - 31 | Reserved |

Bit 0 is the least significant bit of the 32-bit word.

Direction is relative to host / DTE. i.e. for Bluetooth controllers, Send is Host->Controller, Receive is Controller->Host.

Note: Some Datalink Types already encode some or all of this information within the Packet Data. With these Datalink Types, these flags should be treated as informational only, and the value in the Packet Data should take precedence.

Cumulative Drops

A 32-bit unsigned integer representing the number of packets that were lost by the system that created the packet file between the first packet record in the file and this one. Packets may be lost because of insufficient resources in the capturing system, or for other reasons.

Note: some implementations lack the ability to count dropped packets. Those implementations may set the cumulative drops value to zero.

Timestamp Microseconds

A 64-bit signed integer representing the time of packet arrival, in microseconds since midnight, January 1st, 0 AD nominal Gregorian.

In order to avoid leap-day ambiguity in calculations, note that an equivalent epoch may be used of midnight, January 1st 2000 AD, which is represented in this field as 0x00E03AB44A676000.

Packet Data

Variable-length field holding the packet that was captured, beginning with its datalink header. The Datalink Type field of the file header can be used to determine how to decode the datalink header. The length of the Packet Data field is given in the Included Length field.

Note that the length of this field is not necessarily rounded to any particular multi-octet boundary, as might otherwise be suggested by the diagram.

Data Format

All integer values are stored in "big-endian" order, with the high-order bits first.

7.2.3 Ring Indicator

The following information applies when operating the analyzer in **Spy** mode or **Source DTE, No FTS Cables** mode. When using the cables supplied with the analyzer to capture or source data, Ring Indicator (RI) is routed to a different pin which generates interrupts normally.

There is a special case involving Ring Indicator and computers with 8250 UARTs or UARTs from that family where the state of RI may not be captured accurately. Normally when a control signal changes state from high to low or low to high, an interrupt is generated by the UART, and the analyzer goes to see what has changed and record it. Ring Indicator works a little differently. An interrupt is generated when RI changes from high to low, but not when RI changes from low to high. If Ring Indicator changes from low to high, the analyzer does not know that RI has changed state until another event occurs that generates an interrupt. This is simply the way the UART works, and is not a deficiency in the analyzer software.

To minimize the chance of missing a Ring Indicator change, the analyzer polls the UART every millisecond to see if RI has changed. It is still possible for the analyzer to miss a Ring Indicator change if RI and only RI changes state more than once per millisecond.

UARTs in the 8250 family include 8250s, 16450s, 16550s and 16550 variants. If you have any questions about the behavior of your UART and Ring Indicator, please [contact technical support](#).

7.2.4 Progress Bars

The analyzer uses progress bars to indicate the progress of a number of different processes. Some progress bars (such as the filtering progress bar) remain visible, while others are hidden.

The title on the progress bar indicates the process underway.

7.2.5 Event Numbering

This section provides information about how events are numbered when they are first captured and how this affects the display windows in the analyzer. The information in this section applies to frame numbering as well.

When the analyzer captures an event, it gives the event a number. If the event is a data byte event, it receives a byte number in addition to an event number. There are usually more events than bytes, with the result is that a byte might be listed as Event 10 of 16 when viewing all events, and Byte 8 of 11 when viewing only the data bytes.

The numbers assigned to events that are wrapped out of the buffer are not reassigned. In other words, when event number 1 is wrapped out of the buffer, event number 2 is not renumbered to event 1. This means that the first event in the buffer may be listed as event 11520 of 16334, because events 1-11519 have been wrapped out of the buffer. Since row numbers refer to the event numbers, they work the same way. In the above example, the first row would be listed as 2d00 (which is hex for 11520.)

The advantage of not renumbering events is that you can save a portion of a capture file, send it to a colleague, and tell your colleague to look at a particular event. Since the events are not renumbered, your colleague's file use the same event numbers that your file does.

7.2.6 Useful Character Tables

7.2.6.1 ASCII Codes

| hex | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | xA | xB | xC | xD | xE | xF |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|----|----|----|-----|
| 0x | NUL | SOH | STX | ETX | EOT | ENO | ACK | BEL | BS | HT | LF | VT | FF | CR | SO | SI |
| 1x | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM | SUB | ESC | FS | GS | RS | US |
| 2x | SP | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |
| 6x | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w | x | y | z | { | | } | ~ | DEL |

7.2.6.2 Baudot Codes

| DEC | HEX | LETTERS | FIGURES |
|-----|-----|-------------|-------------|
| 0 | 00 | BLANK (NUL) | BLANK (NUL) |
| 1 | 01 | E | 3 |
| 2 | 02 | LF | LF |
| 3 | 03 | A | - |
| 4 | 04 | SP | SP |
| 5 | 05 | S | BEL |
| 6 | 06 | I | 8 |
| 7 | 07 | U | 7 |
| 8 | 08 | CR | CR |
| 9 | 09 | D | \$ |
| 10 | 0A | R | 4 |
| 11 | 0B | J | ' |
| 12 | 0C | N | , |
| 13 | 0D | F | ! |
| 14 | 0E | C | : |
| 15 | 0F | K | (|
| 16 | 10 | T | 5 |
| 17 | 11 | Z | " |
| 18 | 12 | L |) |
| 19 | 13 | W | 2 |
| 20 | 14 | H | # |
| 21 | 15 | Y | 6 |
| 22 | 16 | P | 0 |
| 23 | 17 | Q | 1 |
| 24 | 18 | O | 9 |
| 25 | 19 | B | ? |
| 26 | 1A | G | & |
| 27 | 1B | FIGURES | FIGURES |
| 28 | 1C | M | . |
| 29 | 1D | X | / |
| 30 | 1E | V | ; |
| 31 | 1F | LETTERS | LETTERS |

7.2.6.3 EBCDIC Codes

| hex | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | xA | xB | xC | xD | xE | xF |
|-----|-----|-----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|-----|-----|-----|
| 0x | NUL | SOH | STX | ETX | PF | HT | LC | DEL | | | SMM | VT | FF | CR | SO | SI |
| 1x | DLE | DC1 | DC2 | TM | RES | NL | BS | IL | CAN | EM | CC | CU1 | IFS | IGS | IRS | IUS |
| 2x | DS | SOS | FS | | BYP | LF | ETB | ESC | | | SM | CU2 | | ENQ | ACK | BEL |
| 3x | | | SYN | | PN | RS | UC | EOT | | | | CU3 | DC4 | NAK | | SUB |
| 4x | SP | | | | | | | | | | | . | < | (| + | |
| 5x | & | | | | | | | | | | | \$ | * |) | : | ^ |
| 6x | - | / | | | | | | | | | | . | % | - | > | ? |
| 7x | | | | | | | | | | | : | # | @ | ' | = | " |
| 8x | | a | b | c | d | e | f | g | h | i | | | | | | |
| 9x | | j | k | l | m | n | o | p | q | r | | | | | | |
| Ax | | ~ | s | t | u | v | w | x | y | z | | | | | | |
| Bx | | | | | | | | | | | | | |] | | |
| Cx | { | A | B | C | D | E | F | G | H | I | | | | | | |
| Dx | } | J | K | L | M | N | O | P | Q | R | | | | | | |
| Ex | \ | | S | T | U | V | W | X | Y | Z | | | | | | |
| Fx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | | | |

7.2.6.4 Communication Control Characters

Listed below in alphabetical order are the expanded text meanings for common ANSI communication control characters, and two-character system abbreviation for each one. Some abbreviations have forward slash characters between the two letters. This is to differentiate the abbreviations for a control character from a hex number. For example, the abbreviation for Form Feed is listed as F/F, to differentiate it from the hex number FF.

Table 7.3 - Communications Control Characters

| Abbreviation | Control Character | Text |
|--------------|-------------------|---------------------------|
| AK | ACK | Acknowledge |
| BL | BEL | Bell |
| BS | BS | Backspace |
| CN | CAN | Cancel |
| CR | CR | Carriage Return |
| D/1-4 | DC1-4 | Device Control 1-4 |
| D/E | DEL | Delete |
| DL | DLE | Data Link Escape |
| EM | EM | End of Medium |
| EQ | ENQ | Enquiry |
| ET | EOT | End of Transmission |
| E/C | ESC | Escape |
| E/B | ETB | End of Transmission Block |
| EX | ETX | End of Text |
| F/F | FF | Form Feed |

Table 7.3 - Communications Control Characters(continued)

| Abbreviation | Control Character | Text |
|--------------|-------------------|-----------------------|
| FS | FS | File Separator |
| GS | GS | Group Separator |
| HT | HT | Horizontal Tabulation |
| LF | LF | Line Feed |
| NK | NAK | Negative Acknowledge |
| NU | NUL | Null |
| RS | RS | Record Separator |
| SI | SI | Shift In |
| SO | SO | Shift Out |
| SH | SOH | Start of Heading |
| SX | STX | Start of Text |
| SB | SUB | Substitute |
| SY | SYN | Synchronous Idle |
| US | US | Unit Separator |
| VT | VT | Vertical Tabulation |

7.2.7 DecoderScript Overview

The DecoderScript™ Reference Manual and User Guide is delivered with each Frontline ComProbe® Protocol Analysis System installation package under Developer Tools. The manual is also available on-line at FTE.com.

The main purpose of this manual is to describe DecoderScript™, the language used in writing decoders. DecoderScript allows you to create new decoders or modify existing decoders to expand the functionality of your ComProbe protocol analyzer. DecoderScript displays protocol data, checks the values of fields, validates checksums, converts and combines field values for convenient presentation. Decoders can also be augmented with custom C++-coded functions, called "methods", to extend data formatting, validation, transformations, and so on.

A decoder defines field-by-field how a protocol message can be taken apart and displayed. The core of each "decoder" is a program that defines how the protocol data is broken up into fields and displayed in the Frame Display window of the analyzer software.

This manual provides instruction on how to create and use custom decoders. When reading the manual for the first time, we encourage you to read the chapters in sequence. The chapters are organized in such a way to introduce you to DecoderScript writing step- by- step.

Screenshots of the ComProbe protocol analyzer have been included in the manual to illustrate what you see on your own screen as you develop decoders. But you should be aware for various reasons, the examples may be slightly different from the ones that you create. The differences could be the result of configuration differences or because you are running a newer version of the program. Do not worry if an icon seems to be missing, a font is different, or even if the entire color scheme appears to have changed. The examples are still valid.

Examples of decoders, methods, and frame recognizers are included in this manual. You can cut and paste from these examples to create your own decoders.

A quick note here: Usually the pasted code appears the same as the original in your editor. Some editors, however, change the appearance of the text when it is pasted (something to do with whether it is ASCII or Unicode text). If you find that the pasted text does not appear the same as the original, you can transfer the code into a simple text editor like Notepad, save it as an ANSI (ASCII) file, then use it in your decoder.

These files are installed in the FTE directory of the system Common Files directory. The readme file in the root directory of the protocol analyzer installation contains a complete list of included files. Most files are located in My Decoders and My Methods.

We will be updating our web site with new and updated utilities, etc, on a regular basis and we urge decoder writers to check there occasionally.

7.2.8 Bluetooth low energy ATT Decoder Handle Mapping

Low energy device attributes contain a 16-bit address called the attribute handle. Each handle is associated with an attribute Universally Unique Identifier (UUID) that is 128-bits long. In the attribute database, the handle is unique while the UUID is not unique.

The ComProbe software detects and stores the relationships (mappings) between handle and UUID during the GATT discovery process. But sometimes, there is no GATT discovery process because

- The discovery has previously taken place and both devices stored the mappings and the discovery will not repeat at every subsequent connection.
- The developer owns both devices in the conversation and chose to ignore discovery because the mappings are known.
- The devices are in development and the code to perform the mappings has not been written yet.

The solution to this problem is to

1. define the mappings in a file and
2. then pre-loading the mapping using the ComProbe software.

Creating handle-UUID mapping file

Create a file named "ATT_Handle_UUID_Preload.ini" in the root directory of "C:\Users\Public\Public Documents\Frontline Test Equipment\My Decoders\", but the file can be located anywhere.

Assume that you want to create a GATT service starting at handle 1.

Create a section in the ini file called

```
[Service Base Handles]
A=1
```

"A" will be your first service. Make the base handle equal to the handle of your service. You can use all upper and lower case letters so you can have up to 52 service handles.

Next add the following section.

```
[Advertiser Handles]
; Generic Access Profile (GAP)
A0 = 1800
A1 = 2803
```

A2 = 2a00
A3 = 2803
A4 = 2a01
A5 = 2803
A6 = 2a04

A few things of note:

- In the code above, lines begging with a semi-colon are comments.
- If you want to change the base handle of the GAP service, change the "1" to some other number.
- If you want to comment out the entire service, comment out the base handle. If no "A" is defined, the software will ignore "A1", "A2" and so on.

Contacting Frontline Technical Support

Technical support is available in several ways. The online help system provides answers to many user related questions. Frontline's website has documentation on common problems, as well as software upgrades and utilities to use with our products.

On the Web: <http://fte.com/support/supportrequest.aspx>

Email: tech_support@fte.com

If you need to talk to a technical support representative about your Frontline BPA 600 product, support is available between 9 am and 5 pm, U.S. Eastern Time zone, and between 9 am and 5 pm, Pacific Time zone, on Monday through Friday. Technical support is not available on U.S. national holidays.

Phone: +1 (434) 984-4500

Fax: +1 (434) 984-4505

Instructional Videos

Teledyne LeCroy provides a series of videos to assist the user and may answer your questions. These videos can be accessed at fte.com/support/videos.aspx. On this web page use the **Video Filters** sidebar to select instructional videos for your product.

Appendices

| | |
|-------------------------------------|-----|
| Appendix A: Application Notes | 326 |
|-------------------------------------|-----|

Appendix A: Application Notes

| | |
|--|-----|
| A.1 Audio Expert System: aptX 'hiccup' Detected | 327 |
| A.2 Getting the Android Link Key for Classic Decryption | 334 |
| A.3 Decrypting Encrypted Bluetooth® data with ComProbe BPA 600 | 340 |
| A.4 Decrypting Encrypted Bluetooth® low energy | 348 |
| A.5 Bluetooth® low energy Security | 358 |
| A.6 Bluetooth Virtual Sniffing | 365 |

A.1 Audio Expert System: aptX 'hiccup' Detected

This paper presents a case study in Bluetooth® audio debugging that highlights the importance of Frontline's Audio Expert System (AES) in the process. The actual case involves transmission of a high quality, stereo audio using the aptX codec from a smartphone to a *Bluetooth* headset. The transmission contained SBC encoded packets despite a successful negotiation of aptX encoding and decoding mechanism between the source and the sink devices. Frontline's AES software discovered this transmission error which most likely would not have been easily discovered by using traditional *Bluetooth* protocol and event analysis. Without the Audio Expert System a product may have been shipped that was not performing as expected by the manufacturer.

A.1.1 Background

In *Bluetooth* technology, Audio/Video Distribution Transport Protocol (AVDTP) uses Advanced Audio Distribution Profile (A2DP) for streaming audio in stereo. The A2DP encompasses compression techniques to reduce the amount of radio frequency bandwidth required to transmit audio. In addition to A2DP, Audio/Video Remote Control Profile (AVRCP) controls certain functions of the sending device such as pause, play, next track, etc.

All *Bluetooth* products using A2DP are required to implement audio encoding and decoding using low complexity Sub Band Coding (SBC) that supports up to 345 kb per second bit rate for stereo audio. The SBC codec has some issues though. SBC coding and decoding produces some undesirable artifacts in the audio signal. In addition, the SBC encoding and decoding cycle introduces a time lag in the audio. To improve on SBC's artifacts and time lag issues, a CSR proprietary codec that is called aptX® is implemented on some Bluetooth products.

During the negotiation phase, both Bluetooth devices handshake and they automatically discover the best codec and the highest bit rate to use for audio. If both devices support aptX, it is used rather than the default SBC.

The AES software helps identify audio issues in *Bluetooth* protocol by highlighting information, warnings, and errors related to audio data, codec used, and *Bluetooth* protocol implementation. They are collectively called "events" in AES. The AES window shows audio data plotted as PCM samples versus time in the Wave Panel. The audio data, codec, and protocol events are also graphically displayed in the Wave Panel, and with a single click on an event, engineers and testers are brought directly to the exact packets or frames related to the event in the *Bluetooth* protocol trace in the Frame Display. This helps users find issues quickly and easily. The events are shown time aligned with both the actual audio waveform and bit rate variances graph in the Wave Panel. The bit rate variance graph shows the average or actual amount of Bluetooth audio data sent over a period of time.

AES can operate in two modes: 1) referenced mode, and 2) non-referenced mode. In referenced mode a Frontline provided audio test file is streamed between the Devices Under Test (DUTs). The test file content and parameters are known to the AES software that performs a comparison for deviations. This process helps the software accurately detect anomalies created by the streaming process. In non-referenced mode DUTs stream audio of unknown content, limiting the types of detectable events. The software automatically determines the operation mode with no user input required.

A.1.2 Test Setup

The following DUTs below were used in our test setup:

- DUT1 = smartphone with *Bluetooth* and aptX capability. The smartphone operating system was Android.
- DUT2 = Earphones with *Bluetooth* and aptX capability.

The protocol analyzer: ComProbe BPA 600 Dual Mode *Bluetooth* Protocol Analyzer with *Bluetooth* Audio Expert System activated. The BPA 600 is connected to a personal computer (PC) that is running ComProbe Protocol Analysis System software.

DUT1 was used as a source device. DUT1 was streaming an AES Reference file.

DUT2 was used as a sink device. After establishing a valid *Bluetooth* link, DUT2 played the AES Reference file.

The audio test file was played from the Bluetooth smart phone to the Bluetooth headphone. The data captured by the ComProbe BPA 600 hardware was sent to the analysis computer running ComProbe software with AES. As the data was captured, it was analyzed by the AES module and displayed live in the AES window. The AES software automatically detected the test ID tones in the captured audio and operated in the referenced mode. The figure 1 below shows the test setup.

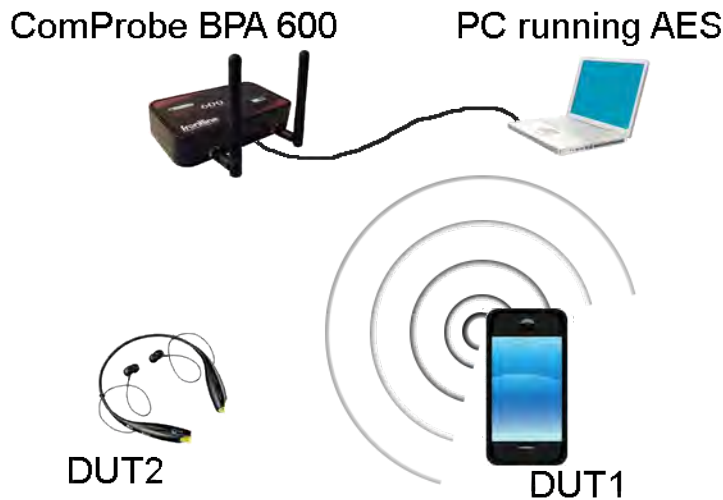


Figure 1 - The Test Setup.

A.1.3 Discussion

The test began without any issue. DUT1 and DUT2 negotiated a Bluetooth connection suitable for transmitting the audio. When the Reference Audio was played there were no obvious audio distortions or anomalies heard by the tester.

The tester used a ComProbe BPA 600 configured for capturing Classic Bluetooth over a single connection.

In Frame Display AVDTP Signaling tab we see the start of the negotiation between DUT1 and DUT2 to establish an audio connection, see Figure 2. At frames 2089 and 2092 the initiating or local device sends an AVDTP_DDISCOVER command. The remote device responds by identifying the ACP Stream Endpoint IDs. In this case the remote device identifies three audio media-type devices that are SNK (sink) devices currently not in use: SEPID (Stream Endpoint Identification) 5, 2, and 1.

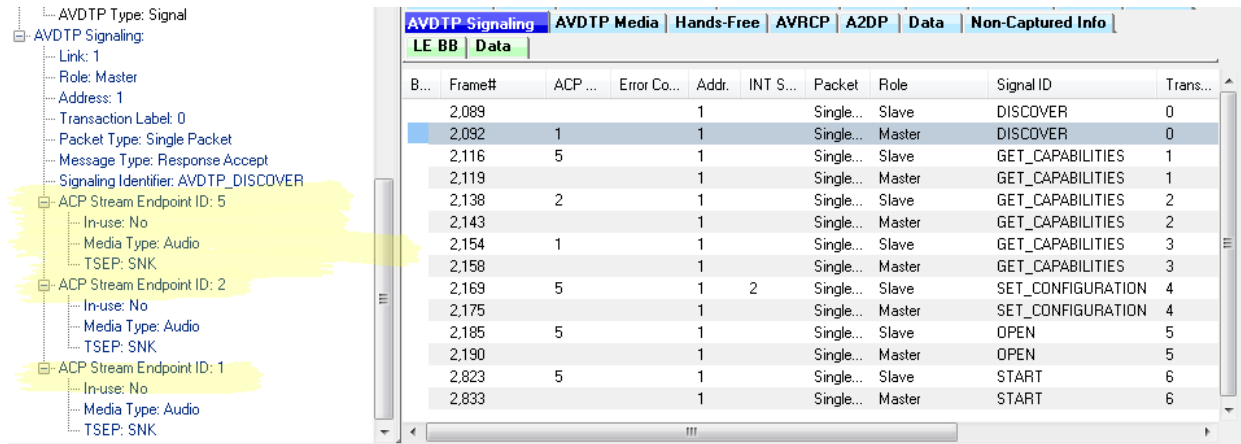


Figure 2 - Frame Display for AVDTP Signaling Frame 2089 & 2092

Note: "ACP" is AVDTP terminology for the remote device.

The next step in the negotiation is to get the audio capabilities of each SEPID. For each SEPID there is an exchange of GET_CAPABILITIES AVDTP signals.

Examination of the Frame Display AVDTP Signaling protocol tab shows at frame 2116 the slave device request SEP (Stream End Point) characteristics. for SEPID (SEP Identifier) 5. Details of the GET_CAPABILITIES command are shown in the Figure 3.

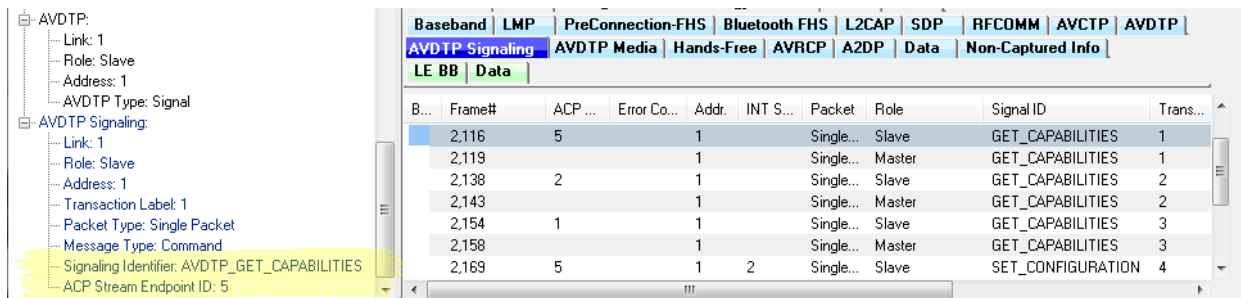


Figure 3 - Frame Display for AVDTP Signaling Frame 2116

At frame 2119 the remote device responds to the GET_CAPABILITIES for SEPID 5 reporting that this SEP codec is aptX with a Channel Mode Stereo.

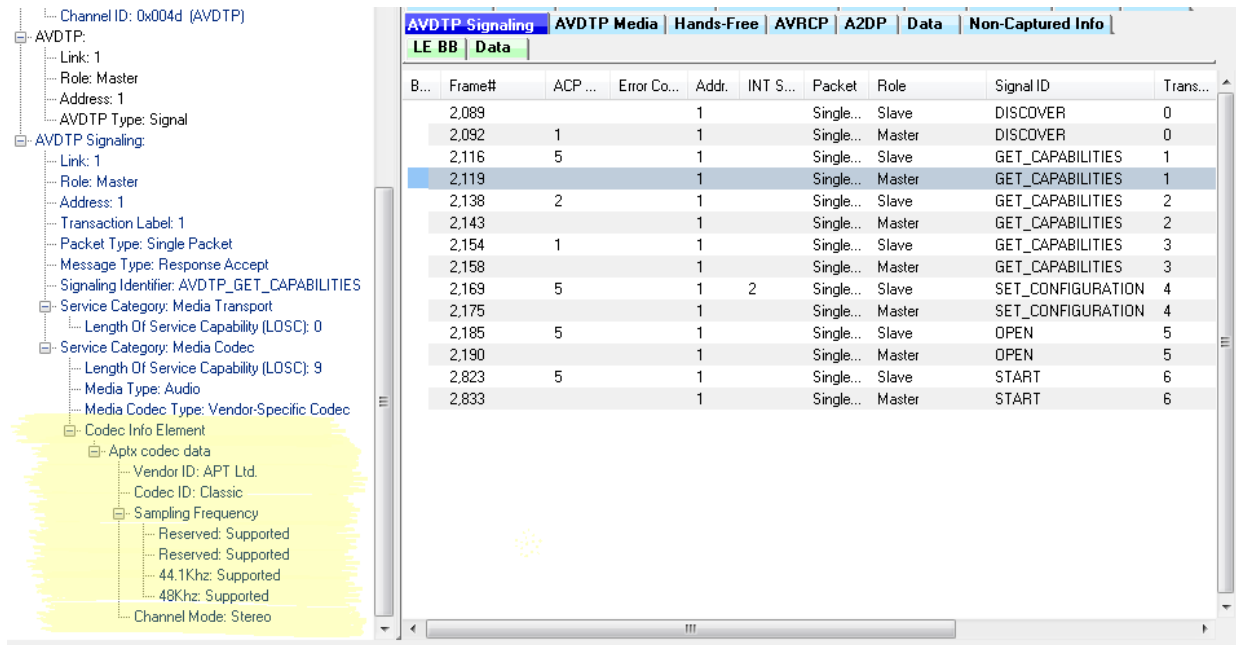
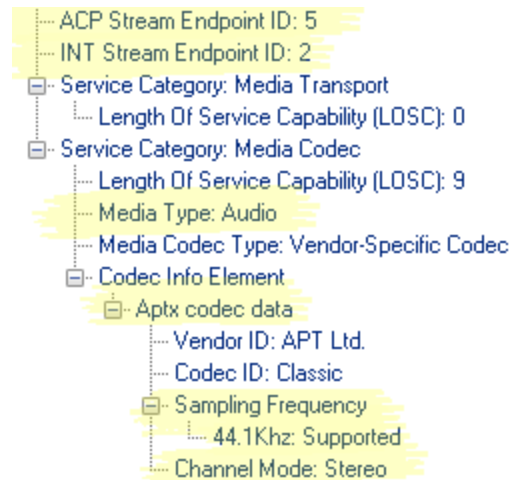


Figure 4 - Frame Display for AVDTP Signaling Frame 2119

In Figure 4, frames 2138 through 2158 perform the GET_CAPABILITIES negotiation between the local and remote device for SEPIDs 2 and 1. SEPID 2 is an MPEG SEP, and SEPID 1 is the SBC SEP.



Frames 2169 and 2175 sets the specific details of the connection with the SET_CONFIGURATION signal. The local device sets the remote endpoint to the aptX device (ACP Stream Endpoint ID: 5), and sets the local endpoint to SEPID 1 (INT Stream Endpoint ID: 2). The Codec, Sampling Frequency, and Channel Mode are also configured. See Figure 5.



At frame 2175 the remote device sends the message "Response Accept" completing the audio stream setup.

Frames 2185 and 2190 are the local request and the remote response to OPEN the audio stream.

Frames 2823 and 2833 START the audio stream with the local request and the remote response respectively.

The Error that is identified by Event 4, the Severity red circle  , is a codec  event at Frame 2839 states "Unable to process AptX data as extracted. It appears that SBC encoded data is being sent over this stream."

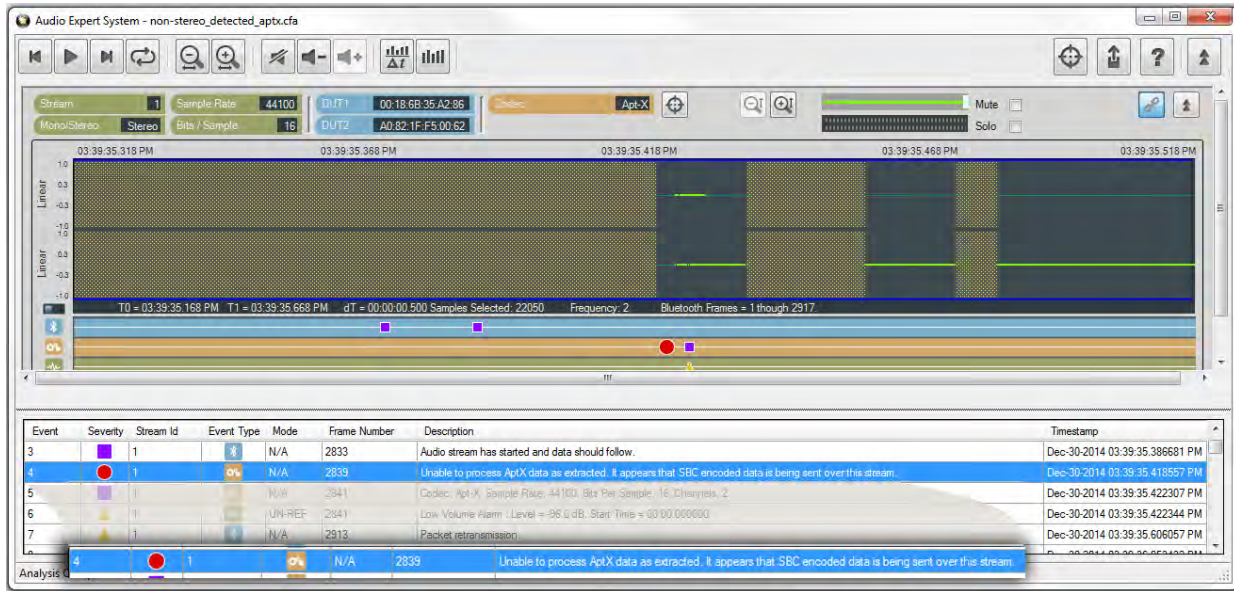


Figure 7 - Audio Expert System Error on Frame 2839: Data not aptX.

A.1.4 Conclusions

This case shows the value of Frontline's Audio Expert System. An error in the transmission of an audio stream compressed using aptX was not easily detected in the protocol analysis using frames. While, in this situation with audio streaming between a smartphone and a *Bluetooth* headset, there was not a significant disruption of the audio, but in playback using other devices there may have been a more significant interruption of the audio streaming.

The smartphone manufacturer may wish to find out why aptX compressed audio contained SBC compressed data in the stream. We can speculate that there may be an underlying problem with clearing stacks or memory between streaming events. This investigation is beyond the scope of this paper.

If there is interest in the Audio Expert System as an expansion of your ComProbe Bluetooth analyzer contact the Frontline sales at sales@fte.com or visit our web site at fte.com.

Author: John Trinkle & Priyanka Gupta

Publish Date: 27 February 2015

A.2 Getting the Android Link Key for Classic Decryption

Bluetooth devices on an encrypted link share a common “link key” used to exchange encrypted data. For a *Bluetooth* sniffer, such as the ComProbe BPA 600, to be able to decrypt the encrypted data, it must also have this shared link key. For obvious security reasons, the link key is never sent over the air, so either the user must get the key out of one of the devices being sniffed and supply the key to the sniffer or the sniffer must create the key itself.

Bluetooth devices using the Android operating system have a "developer" option that will provide the link key for Classic *Bluetooth* decryption. This procedure will use the developer options to obtain the Android HCI (Host Controller Interface) log that contains the link keys for all active links..

A.2.1 What You Need to Get the Android Link Key

The process applies to the Android 4.4 or later operating system.

- Android device with Bluetooth enabled and paired with another *Bluetooth* device.
- ComProbe Protocol Analysis System installed on your computer
- Android Debug Bridge (optional)

Note: Each Android device model can vary in screen organization, layout and format. The directions in this paper are based on known typical Android device. Refer to the manufacturer’s manual, on-line help, or technical support for detailed information about your particular device.

A.2.2 Activating Developer options

The Android HCI log will contain the link key for an active *Bluetooth* link.

1. On the Android device go to **Settings**,
2. Select **About**.
3. In the About screen tap on **Build number** eight times. At some point you will see a notice similar to "You

are now a developer!".

Note: On some devices the build information may be under one or more sub-screens below the About screen. Also the number of taps may vary; in most cases the screen will provide status of your tap count.

4. Return to the **Settings** screen and you will see **Developer options**

A.2.3 Retrieving the HCI Log

Now that **Developer options** have been activated on the Android device, you can retrieve the HCI log.

1. On the Android device go to **Settings**.
2. Select **Developer options**.
3. Click to enable **Bluetooth HCI snoop logging**.
4. Return to the **Settings** screen and select **Developer options**.
5. In the **Developer options** screen select **Enable Bluetooth HCI snoop log**. The log file is now enabled.

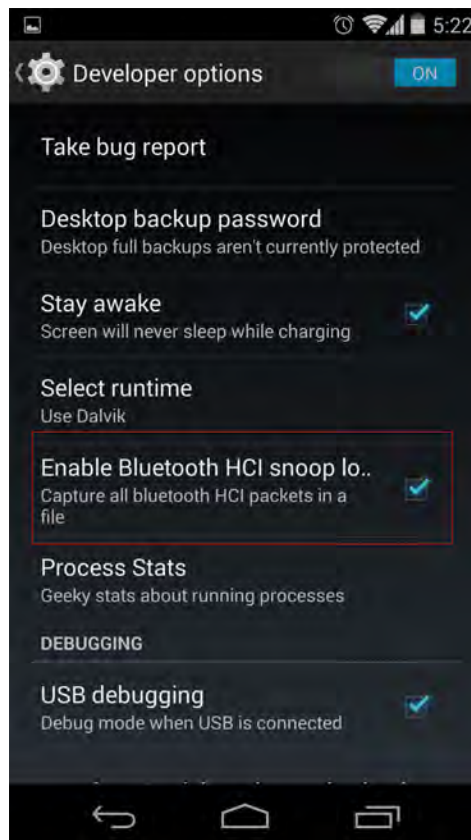


Figure 8 - Typical Android Developer options screen

6. On the Android device turn off *Bluetooth*.
7. Turn on *Bluetooth*.
8. Reboot the Android device.

The HCI log file is now being generated and is saved to */sdcard/btsnoop_hci.log*.

Note: Samsung devices have a slightly different location for the btsnoop file.

There are two options for retrieving the HCI log from the Android device.

- a. Attach the Android device to your computer. The file */sdcard/btsnoop_hci.log* is in the root of one of the mountable drives. Copy the file to directory *C:/Users/Public/Public Documents/Frontline Test Equipement/My Capture File/*.
- b. The second option is to use the Android Debug Bridge (ADB) using the following steps. The debug bridge is included with Android Software Developer Kit.

- (1). On the Android device **Development** screen, select **Android debugging** or **USB debugging**.
- (2). Connect your computer and Android device with a USB cable.
- (3). Open a terminal on your computer and run the following command.

```
adb devices.
```

- (4). Your Android device should show up in this list confirming that ADB is working.

```
List of devices attached  
XXXXXXXXXXXX device
```

- (5). In the terminal enter the following command to copy the HCI Log to your computer.

```
adb pull /sdcard/btsnoop_hci.log
```

A.2.4 Using the ComProbe Software to Get the Link Key

You will load the HCI Log file *btsnoop_HCI.log* into the ComProbe Protocol Analysis System on your computer as a capture file. Then you can use the **Frame Display** to locate the link key.

1. Activate the ComProbe Protocol Analysis System. (Refer to the ComProbe BPA 600 User Manual on fte.com).
2. From the Control window menu select **File, Open Capture File...**
3. When the **Open** window appears, set the file type to **BTSnoop Files (*.log)**. If not already selected navigate to the *My Capture Files* directory and select *btsnoop_hci.log*.

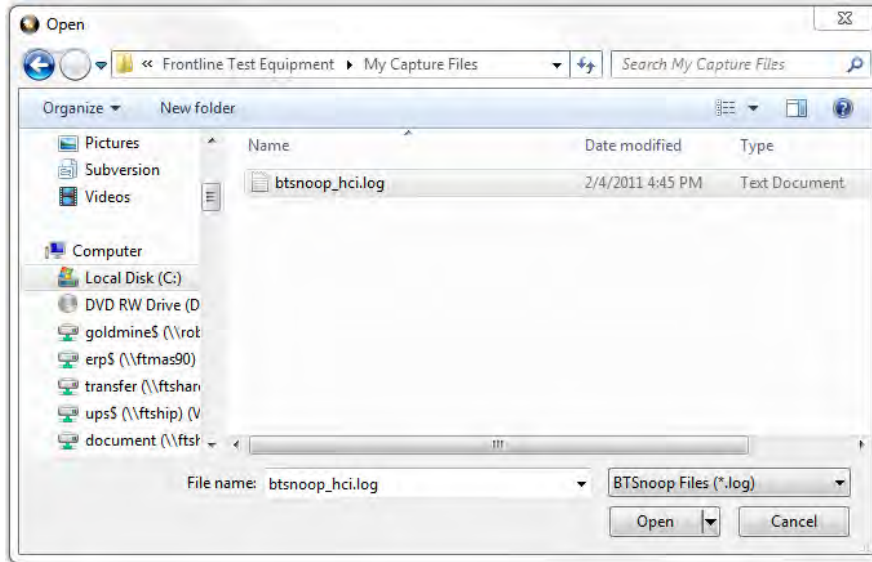




Figure 9 - Select Capture File

4. Open the **Frame Display** 
5. In the **Frame Display** protocol tabs select **HCI**. (See image below)
6. Select Find  , click on the **Decode** tab, and enter "link key" in the Search for String in Decode. Check the **Ignore Case** option. Click on **Find Next** until the Event column shows Link Key Notification.

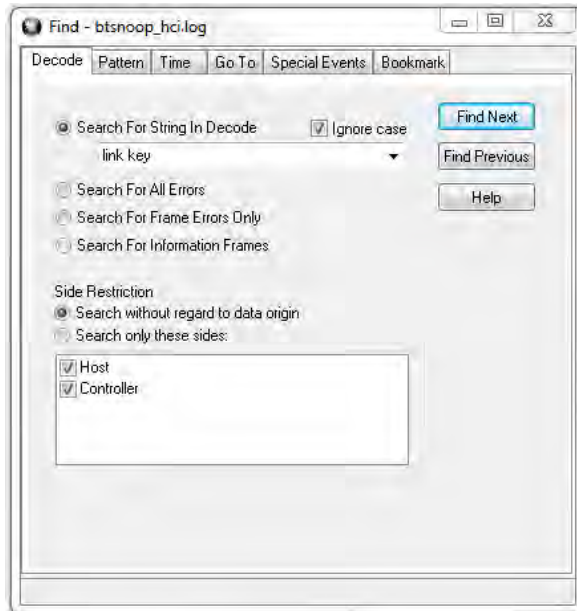


Figure 10 - Find Dialog

In the **Frame Display** Detail pane, expand HCI and HCI Event where the Link Key is shown. Copy and paste the Link Key into the appropriate BPA 600 datasource dialog. (See the example below)

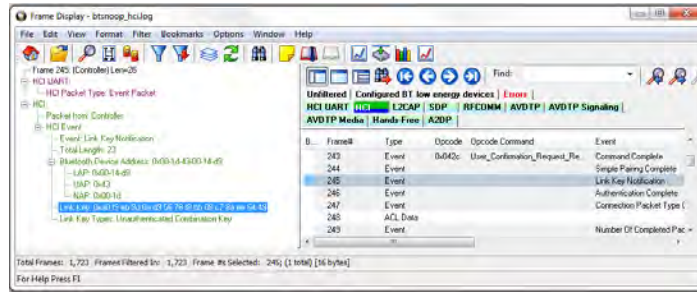


Figure 11 - Frame Display Showing Link Key Notification Event with the Link Key

Author: John Trinkle with Joe Skupniewitz

Publish Date: 30 September 2014

A.3 Decrypting Encrypted Bluetooth® data with ComProbe BPA 600

A.3.1 How Encryption Works in *Bluetooth*

Bluetooth devices on an encrypted link share a common “link key” used to exchange encrypted data. How that link key is created depends on the pairing method. Pairing methods have evolved and changed throughout *Bluetooth* history. The earlier legacy method was used up through *Bluetooth 2.0*. Improved and simpler pairing methods began with *Bluetooth 2.1* and remain in the current version *Bluetooth 4.0*.

For a *Bluetooth* sniffer to be able to decrypt the encrypted data, it must also have this shared link key. For obvious security reasons, the link key is never sent over the air, so either the user must get the key out of one of the devices being sniffed and supply the key to the sniffer or the sniffer must create the key itself.

A.3.2 Legacy Pairing (*Bluetooth 2.0* and earlier)

In legacy pairing, this link key is derived from a shared PIN code, the master’s *Bluetooth* clock, the master’s BD_ADDR and a random number that is passed between the two devices. If the sniffer has all of this same data, it can create the link key in the same way that the devices do. The sequence of events used to create this key, or pairing process, is shown in the ComProbe software Frame Display below.

| AVDTP Signaling | | | | AVDTP Media | | |
|-----------------|----------|---------------------------|----------------------|----------------------|---------------|--------------|
| Unfiltered | Baseband | Extended Inquiry Response | | LMP | Bluetooth FHS | |
| B... | Frame# | LT_Addr | Original Opcode | Opcode | Role | Initiated by |
| ● | 246 | 1 | | in_rand | Slave | slave |
| ● | 247 | 1 | | in_rand | Master | master |
| ● | 249 | 1 | in_rand | accepted | Slave | master |
| ● | 250 | 1 | | comb_key | Master | master |
| ● | 251 | 1 | | comb_key | Slave | master |
| ● | 252 | 1 | | au_rand | Master | master |
| ● | 253 | 1 | | sres | Slave | master |
| ● | 254 | 1 | | au_rand | Slave | master |
| ● | 255 | 1 | | sres | Master | master |
| ● | 256 | 1 | | setup_complete | Master | master |
| ● | 257 | 1 | | encrypt_mode_req | Slave | slave |
| ● | 258 | 1 | encrypt_mode_req | accepted | Master | slave |
| ● | 259 | 1 | | encrypt_key_size_req | Master | slave |
| ● | 260 | 1 | encrypt_key_size_req | accepted | Slave | slave |
| ● | 261 | 1 | | start_encrypt_req | Master | slave |

Figure 12 - Frame Display

Frame 247 is the LMP_in_rand which is where a random number generated by the master is passed to the slave. The slave acknowledges that it has accepted the number in frame 249. The initialization key has been passed to the slave and is now shared by both devices. Both devices now independently generate combination keys.

In frames 250 and 251, the combination keys are passed between master and slave. In frame 252, the master sends its LMP_au_rand. This is the random number that has been encrypted using the link key that master has calculated. The slave then responds with frame 253, an LMP_sres confirming that it was able to compute the same number. That process is repeated in the other direction (slave to master) in frames 254 and 255. This completes the authentication between devices, and the setup_complete message is sent and the slave requests encryption mode in frame 257, and the master accepts in frame 258. The actual encryption starts after the start encryption request in frame 261.

In order for the ComProbe software to decrypt an encrypted *Bluetooth* conversation, the ComProbe software must compute the same link key being used by the devices being sniffed. Since this link key is never sent over the air, the ComProbe software must have all of the same information the devices being sniffed have so that it can calculate the same link key that each of the two devices does. To decrypt successfully, the ComProbe software must know the PIN code and capture:

- The LMP_in_rand
- Both LMP_comb_keys
- Both LMP_au_rand/LMP_sres pairs.

If any of these are missed, the ComProbe software will not be able to decrypt. If you capture encrypted data and find that everything captured after the LMP_start_encryption_request is in error, look back at the LMP frames previous to that and you'll probably find one or more of these missing. The Start Encryption Request will also be marked by the ComProbe software with an error that indicates that the link key calculated by the ComProbe software is different from the one used by your devices.

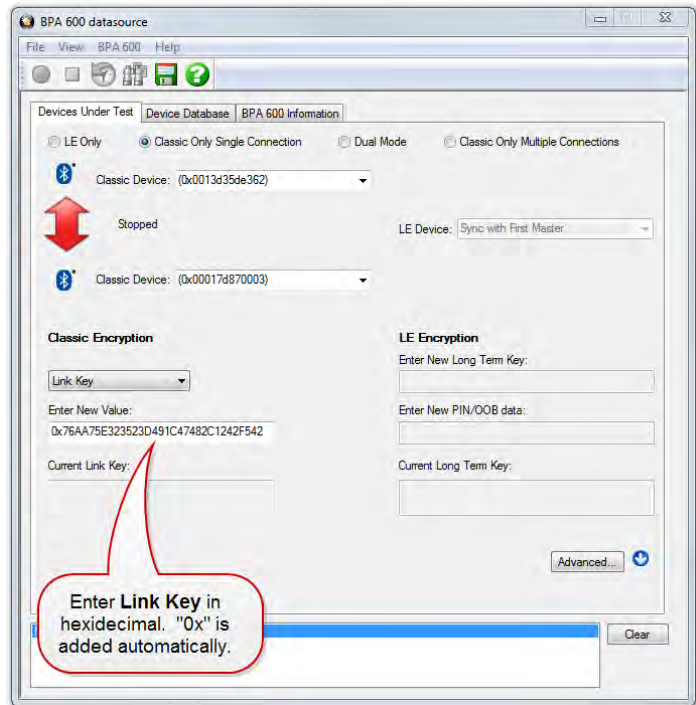
A.3.3 Secure Simple Pairing (SSP) (Bluetooth 2.1 and later)

To capture and decrypt data between two *Bluetooth* devices using Secure Simple Pairing we have two choices. If one of your devices can be put into Secure Simple Pairing Debug Mode, all that needs to be done in I/O Settings is to choose your devices. It doesn't matter what's been selected in the Pairing Method drop down, the ComProbe software will see the debug messages being sent and calculate the correct key. Only one of the devices needs to be in debug mode and it doesn't matter which one.

If neither of your devices can be put into debug mode, you'll need to know the link key being used by one of your devices, generally by accessing the HCI on one of the devices. If that is the case, enter the link key into the box provided.

Note that the link key is sometimes stored in your device in reverse order. The ComProbe software will automatically reverse the link key, if needed.


Once the link key has been entered, decryption operates the same way it does in legacy pairing.




A.3.4 How to Capture and Decrypt Data (Legacy Pairing)


Run the ComProbe software and select **Bluetooth Classic/low energy (BPA 600)**. This will open the **Control** window and the **BPA 600 Datasource** where ComProbe device parameters are set for sniffing including the devices to be sniffed and how the link key is to be encrypted.


Select the **Devices Under Test** tab. Make both your *Bluetooth* devices discoverable.



Click the **Discover Devices**  on the datasource toolbar. The ComProbe software will find any discoverable *Bluetooth* devices within its range. You will then be able to select your devices from the drop down lists. If one or both of your devices cannot be made discoverable, you may type in the BD_ADDR(s) directly.

With legacy pairing, select **PIN Code (ASCII)** from the **Classic Encryption** drop down and fill in the PIN. As mentioned above, the ComProbe software needs the PIN code in order to calculate the link key the two *Bluetooth* devices are using. Alternately, you may enter the Link Key manually if it is known. The ComProbe software also keeps a database of the link keys it previously calculated, which may be accessed on the **Device Database** tab.

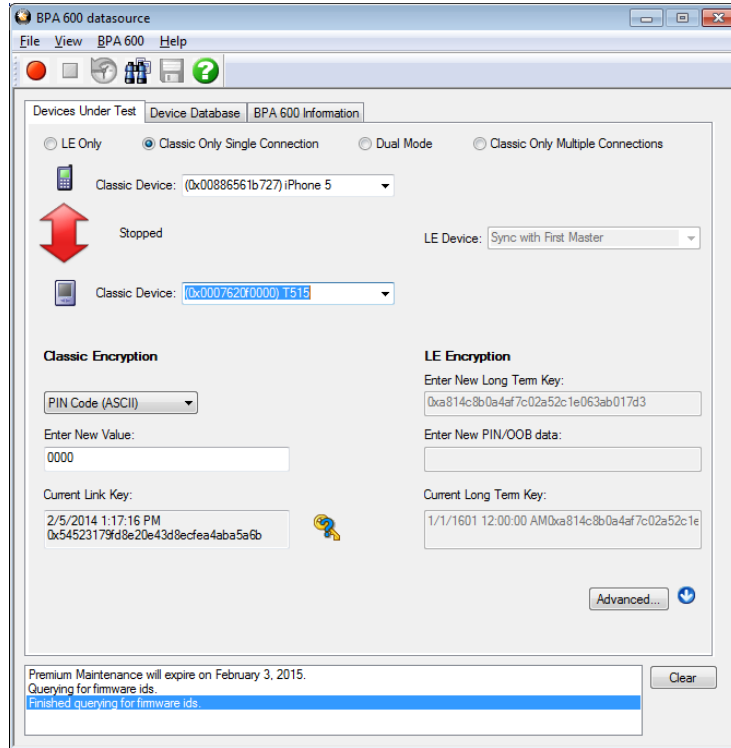
The **Start Sniffing** button  should now be available. If Start Sniffing is grayed out, there is something set up incorrectly in the datasource **Device Under Test** tab. For example, if you selected PIN code in the encryption drop down but you neglected to fill in the PIN code, then Start Sniffing will be grayed out.

Click on the toolbar **Start Sniffing** button. The **Control** window will display a capture status message. When you start sniffing, the colored arrow be red indicating that the Bluetooth devices are initializing. . After a few seconds the arrow will turn green  and the status will change to “Waiting for the master to connect to the slave”. At this point the BPA 600 is synchronized and waiting for a baseband connection.

When your connection is established, the arrow will turn blue , signifying that a baseband link has been established and data should start to appear in the **Frame Display**. The direction of the arrow indicates which device is master and which is slave. The arrow points from master to slave.

If ComProbe software successfully calculates the correct link key, the Link Key icon  on the datasource is updated with a check mark to indicate that the link key has been verified. Should the link key be incorrect the link key icon will show .

An incorrect link key will show up in the **Frame Display**. Open the **Frame Display LMP** tab and search for frames with errors appearing in red. In the **Decode** pane a link key error will appear in red under **Errors**.



Frame 14,382: (Master) Len=29

- Errors:
 - Link Key Error - The Link Key used by FTS is not the same key that the pair of devices Authenticated.
 - LMP - Link Key Error: The Link Key used by FTS is not the same key that the pair of devices Authenticated. [=0]
- Baseband:
 - Header Length: 11
 - Header Version: 3
 - Link: 1
 - Role: Master (0x07-62-0f-00-00-00) (#1)
 - Channel: 59 - 2461 MHz
 - Clock: 0x0003fec
 - Packet Status: OK

A.3.5 How to tell if a device is in Secure Simple Pairing Debug Mode

When a device is configured in SSP debug mode, the ComProbe software will decode and display the debug key in the Encapsulated Payload message of the **Frame Display Summary** pane. There will be an Encapsulated Payload message sent from both the master and the slave. The message from the device that is in debug mode will show the debug key, the other will show the public key. Refer to the **Frame Display Decode** pane in the screenshots below where the master is in SSP debug mode. Remember, only one of the *Bluetooth* devices needs to be in SSP debug mode.

| Unfiltered | | Non-Captured Info | | Errors | | Info | | |
|------------|--------|-------------------|----------------------|------------------------|----------|--------------|---------|-----------------|
| Baseband | LMP | Bluetooth FHS | SCO/eSCO | L2CAP | SDP | RFCOMM | AVDTP | AVDTP Signaling |
| B... | Frame# | LT_Addr | Original Opcode | Opcode | Role | Initiated by | Fram... | |
| ● | 393 | 3 | encapsulated_header | accepted | Slave | master | 11 | |
| ● | 396 | 3 | | encapsulated_payload | Master | master | 26 | |
| ● | 407 | 3 | encapsulated_payload | accepted | Slave | master | 11 | |
| ● | 410 | 3 | | encapsulated_payload | Master | master | 26 | |
| ● | 415 | 3 | encapsulated_payload | accepted | Slave | master | 11 | |
| ● | 418 | * 3 | | * encapsulated_payload | * Master | * master | 26 | |
| ● | 423 | 3 | encapsulated_payload | accepted | Slave | master | 11 | |
| ● | 505 | 3 | | preferred_rate | Slave | slave | 11 | |
| ● | 547 | 3 | | encapsulated_header | Slave | master | 13 | |

Frame 418: (Master) Len=26

- * means that the data were reconstructed.
- Baseband:
 - LMP:
 - * Role: Master
 - * Address: 3
 - * Opcode: LMP_encapsulated_payload
 - * Transaction ID: Initiated by master
 - * P-192 Public Key
 - Debug Key(X): 0x 15 20 70 09 98 44 21 a6 58 6f 9f c3 fe 7e 43 29 d2 8
 - * Debug Key(Y): 0x b0 9d 42 b8 1b c5 bd 00 9f 79 e4 b5 9d bb aa 85 7

Figure 13 - Encapsulated Payload Message from a *Bluetooth* Device in SSP Debug Mode

| Unfiltered | Non-Captured Info | Errors | Info | | | | | |
|------------|-------------------|---------------|------------------------|------------------------|---------|--------------|-------|-----------------|
| Baseband | LMP | Bluetooth FHS | SCO/eSCO | L2CAP | SDP | RFCOMM | AVDTP | AVDTP Signaling |
| B... | Frame# | LT_Addr | Original Opcode | Opcode | Role | Initiated by | Fram. | |
| ● | 550 | 3 | encapsulated_header | accepted | Master | master | 11 | |
| ● | 553 | 3 | encapsulated_payload | accepted | Slave | master | 26 | |
| ● | 556 | 3 | encapsulated_payload | accepted | Master | master | 11 | |
| ● | 561 | 3 | encapsulated_payload | accepted | Slave | master | 26 | |
| ● | 564 | 3 | encapsulated_payload | accepted | Master | master | 11 | |
| ● | 571 | * 3 | * encapsulated_payload | * encapsulated_payload | * Slave | * master | 26 | |
| ● | 574 | 3 | encapsulated_payload | accepted | Master | master | 11 | |
| ● | 599 | 3 | Simple_Pairing_Confirm | Simple_Pairing_Confirm | Slave | master | 26 | |
| ● | 602 | 3 | Simple_Pairing_Number | Simple_Pairing_Number | Master | master | 26 | |

Frame 571: (Slave) Len=26

* means that the data were reconstructed.

- Baseband:
- LMP:
 - * Role: Slave
 - * Address: 3
 - * Opcode: LMP_encapsulated_payload
 - * Transaction ID: Initiated by master
 - * P-192 Public Key
 - * X co-ordinate: 0x c2 e2 b5 92 01 e7 e0 53 df 1f d1 40 cd 8f df da df 0c
 - * Y co-ordinate: 0x 9a 39 62 d9 6e 07 e6 fb 36 06 49 52 11 6a a0 e6 e2

```

B 0 1 1 1 1 1 0 0 0 0 1 0 0 0
N 0 1 1 0 0 1 1 0 1 1 0 1 1 1
A 1 1 0 1 1 1 1 1 1 1 0 1 1 0
R 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0

R 7 c 2 1 d 0 6 e 6 6
A c d 4 0 d 1 1 f d f
D c 2 f 3 e c c a 5 8
I 1 1 5 2 4 9 0 6 3 6
X P 9 a
A N E
C H A R A C T E R S
  
```

Figure 14 - Encapsulated Payload Message from a Bluetooth Device NOT in SSP Debug Mode

Author: Sean Clinchy

Publish Date: February 2014

A.4 Decrypting Encrypted Bluetooth® low energy

A.4.1 How Encryption Works in *Bluetooth* low energy

Data encryption is used to prevent passive and active—man-in-the-middle (MITM) — eavesdropping attacks on a *Bluetooth* low energy link. Encryption is the means to make the data unintelligible to all but the *Bluetooth* master and slave devices forming a link. Eavesdropping attacks are directed on the over-the-air transmissions between the *Bluetooth* low energy devices, so data encryption is accomplished prior to transmission using a shared, secret key.

A.4.2 Pairing

A *Bluetooth* low energy device that wants to share secure data with another device must first pair with that device. The Security Manager Protocol (SMP) carries out the pairing in three phases.

1. The two connected *Bluetooth* low energy devices announce their input and output capabilities and from that information determine a suitable method for phase 2.
2. The purpose of this phase is to generate the Short Term Key (STK) used in the third phase to secure key distribution. The devices agree on a Temporary Key (TK) that along with some random numbers creates the STK.
3. In this phase each device may distribute to the other device up to three keys:
 - a. the Long Term Key (LTK) used for Link Layer encryption and authentication,
 - b. the Connection Signature Resolving Key (CSRK) used for data signing at the ATT layer, and
 - c. the Identity Resolving Key (IRK) used to generate a private address.

Of primary interest in this paper is the LTK. CSRK and IRK are covered briefly at the end.

Bluetooth low energy uses the same pairing process as Classic *Bluetooth*: Secure Simple Pairing (SSP). During SSP initially each device determines its capability for input and output (IO). The input can be None, Yes/No, or Keyboard with Keyboard having the ability to input a number. The output can be either None or Display with Display having the ability to display a 6-digit number. For each device in a pairing link the IO capability determines their ability to create encryption shared secret keys.

The Pairing Request message is transmitted from the initiator containing the IO capabilities, authentication data availability, authentication requirements, key size requirements, and other data. A Pairing Response message is transmitted from the responder and contains much of the same information as the initiators Pairing Request message thus confirming that a pairing is successfully negotiated.

In the sample SMP decode, in the figure at the right, note the “keys” identified. Creating a shared, secret key is an evolutionary process that involves several intermediary keys. The resulting keys include,

1. IRK: 128-bit key used to generate and resolve random address.
2. CSRK: 128-bit key used to sign data and verify signatures on the receiving device.
3. LTK: 128-bit key used to generate the session key for an encrypted connection.
4. Encrypted Diversifier (EDIV): 16-bit stored value used to identify the LTK. A new EDIV is generated each time a new LTK is distributed.
5. Random Number (RAND): 64-bit stored value used to identify the LTK. A new RAND is generated each time a unique LTK is distributed.

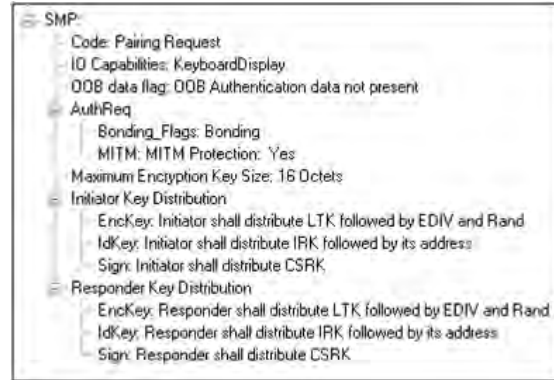


Figure 15 - Sample Initiator Pairing Request Decode (ComProbe Frame Display, BPA 600 low energy capture)

Of particular importance to decrypting the encrypted data on a *Bluetooth* low energy link is LTK, EDIV, and RAND.

A.4.3 Pairing Methods

The two devices in the link use the IO capabilities from Pairing Request and Pairing Response packet data to determine which of two pairing methods to use for generation of the Temporary Key (TK). The two methods are **Just Works** and **Passkey Entry**¹. An example of when **Just Works** method is appropriate is when the IO capability input = None and output = None. An example of when **Passkey Entry** would be appropriate would be if input= Keyboard and output = Display. There are 25 combinations that result in 13 **Just Works** methods and 12 **Passkey Entry** methods.

In **Just Works** the TK = 0. In the **Passkey Entry** method,

$$TK = \begin{cases} 6 \text{ numeric digits, Input = Keyboard} \\ 6 \text{ random digits, Input = Display} \end{cases}$$

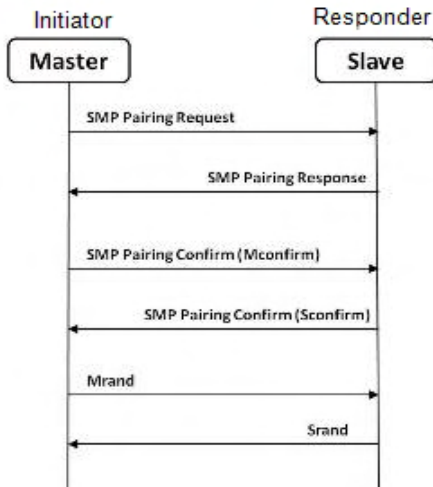


Figure 16 - Initiator Pairing Confirm Example (ComProbe Frame Display, BPA 600 low energy capture)

¹A third method, Out Of Band (OOB), performs the same as **Pass Key**, but through another external link such as NFC.



Figure 17 - Responder Pairing Confirm Example (ComProbe Frame Display, BPA 600 low energy capture)



The initiating device will generate a 128-bit random number that is combined with TK, the Pairing Request command, the Pairing Response command, the initiating device address and address type, and the responding device address and address type. The resulting value is a random number **Mconfirm** that is sent to the responding device by the Pairing Confirm command. The responding device will validate the responding device data in the Pairing Confirm command and if it is correct will generate a **Sconfirm** value using the same methods as used to generate **Mconfirm** only with different 128-bit random number and TK. The responding device will send a Pairing Confirm command to the initiator and if accepted the authentication process is complete. The random number in the **Mconfirm** and **Sconfirm** data is **Mrand** and **Srand** respectively. **Mrand** and **Srand** have a key role in setting encrypting the link.

Finally the master and slave devices exchange **Mrand** and **Srand** so that the slave can calculate and verify Mconfirm and the master can likewise calculate and verify Sconfirm.

Figure 18 - Message Sequence Chart: SMP Pairing

A.4.4 Encrypting the Link

The Short Term Key (STK) is used for encrypting the link the first time the two devices pair. STK remains in each device on the link and is not transmitted between devices. STK is formed by combining **Mrand** and **Srand** which were formed using device information and TKs exchanged with Pairing Confirmation (**Pairing Confirm**).

A.4.5 Encryption Key Generation and Distribution

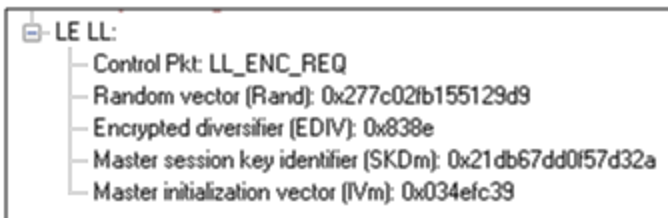


Figure 19 - Encryption Request from Master, Example (ComProbe Frame Display, BPA 600 low energy capture)

To distribute the LTK, EDIV, and Rand values an encrypted session needs to be set up. The initiator will use STK to enable encryption on the link. Once an encrypted link is set up, the LTK is distributed. LTK is a 128-bit random number that the slave device will generate along with EDIV and Rand. Both the master and slave devices can distribute these numbers, but *Bluetooth* low energy is designed to conserve energy, so the slave device is often resource constrained and does not have the database storage resources for holding LTKs. Therefore the slave will distribute

LTK, EDIV, and Rand to the master device for storage. When a slave begins a new encrypted session with a previously linked master device, it will request distribution of EDIV and Rand and will regenerate LTK.

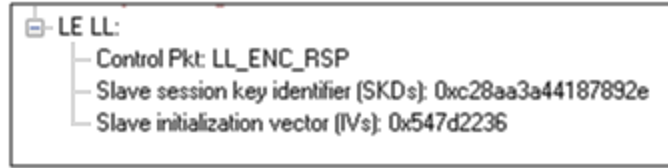


Figure 20 - Encryption Response from Slave, Example
(ComProbe Frame Display, BPA 600 low energy capture)

A.4.6 Encrypting The Data Transmission

Data encryption begins with encrypting the link. The Session Key (SK) is created using a session key diversifier (SKD). The first step in creating a SK is for the master device to send Link Layer encryption request message (LL_ENC_REQ) that contains the SKD_{master}. The SKD_{master} is generated using the LTK. The slave receives SKD_{master}, generates SKD_{slave}, and generates SK by concatenating parts of SKD_{master} and SKD_{slave}. The slave device responds with an encryption response message (LL_ENC_RSP) that contains SKD_{slave}; the master will create the same SK.

Now that a SK has been calculated, the master and slave devices will now begin a handshake process. The slave will transmit unencrypted LL_START_ENC_REQ, but sets the slave to receive encrypted data using the recently calculated SK. The master responds with encrypted LL_START_ENC_RSP that uses the same SK just calculated and setting the master to receive encrypted data. Once the slave receives the master's encrypted LL_START_ENC_RSP message and responds with an encrypted LL_START_ENC_RSP message the *Bluetooth* low energy devices can now begin transmitting and receiving encrypted data.

A.4.7 Decrypting Encrypted Data Using Frontline® BPA 600 low energy Capture

Note: The following discussion uses the ComProbe BPA 600 in low energy capture mode to illustrate how to identify the encryption process and to view decrypted data. However any of the ComProbe devices (BPA 500, BPA low energy) that are low energy capable will accomplish the same objectives, although the datasource setup will be slightly different for each device.

A.4.7.1 Setting up the BPA 600

1. Run the ComProbe Protocol Analysis Software and select **Bluetooth Classic/low energy (BPA 600)**. This will bring up the **BPA 600 datasource** window. This is where the parameters are set for sniffing, including the devices to be sniffed and how the link is to be decrypted.
2. Select **Devices Under Test** tab on the Datasource window.
3. Click/select **LE Only**.
4. To decrypt encrypted data transmissions between the *Bluetooth* low energy devices the ComProbe analyzer needs to know the LTK because this is the shared secret used to encrypt the session. There are two ways to provide this information and which to select will depend on the pairing method: **Just Works** or **Passkey Entry**.

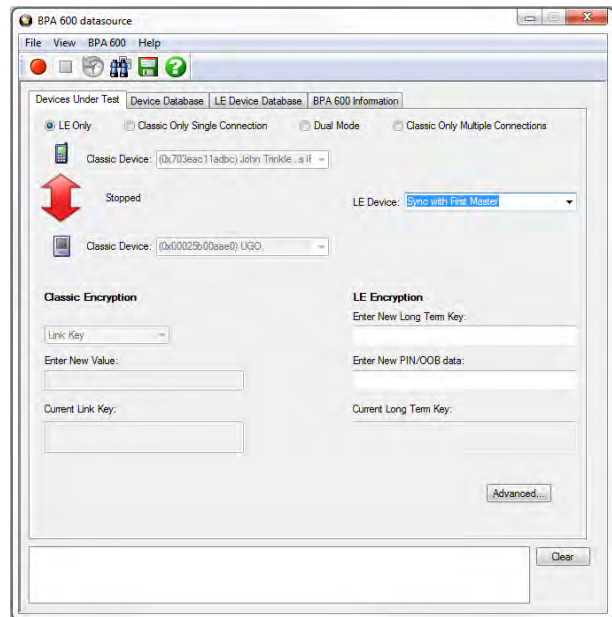


Figure 21 - ComProbe BPA 600 low energy only datasource settings

- a. **Passkey Entry** is easiest if you have the code that was displayed or entered during device pairing. The code is what is used to generate the LTK. Under **LE Encryption** enter the code in the **Enter New PIN/OOB** data text box.
- b. **Just Works** is more of a challenge because you must know the LTK that is created at the time of pairing and identification of an encrypted link.
 - If your device was previously used in an encrypted capture session, the device information including LTK can be found in the **Device Database** tab.
 - In a design and development environment the LTK is often known beforehand.
 - Capture of Host Controller Interface (HCI) events using ComProbe HSU can reveal the LTK, which is contained in the HCI_Link_Key_Request_Reply command. HCI capture is through direct connection to the device host controller. The information obtained in a direct connection can later be used in a wireless encrypted capture session that requires prior knowledge of encryption keys.

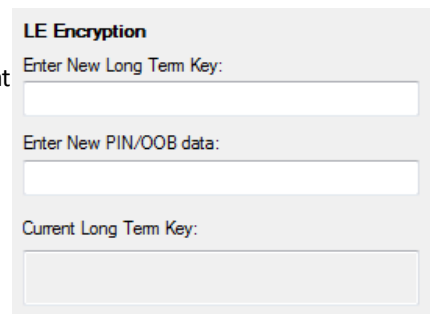



Figure 22 - BPA 600 datasource Encryption Key Entry

5. To start capture click on the Start Sniffing button  on the **BPA 600 datasource** toolbar.

A.4.7.2 Use Frame Display to View Encryption/Decryption Process

A.4.7.2.1 Security Manager Protocol

The Security Manager Protocol (SMP) controls the process for pairing and key distribution. The results of a pairing and key distribution can be observed in the ComProbe software **Frame Display**. Activate the **Frame Display** by clicking on the icon on the **Control** window toolbar. On the **Frame Display** low energy protocols are shown in light green tabs. Click on the **SMP** protocol tab that will show only the SMP commands from the full data set.



Figure 23 - SMP Pairing Request (Frame# 35,539) from Initiator (Side 1)

On the left side of the figure above is the **Frame Display Decoder** pane that shows the decoded information supplied in the selected frame in the Summary pane, Frame# 35,539. Shown is the SMP data associated with and encrypted link (MITM Protection = Yes). The requested keys are also shown. Selecting Frame# 35,545 would provide the response from the responder (Side 2) and would contain similar information.

Selecting Frame# 39,591 will display the Pairing Confirm from the initiator (Side 1) in the **Decoder** pane. The Confirm Value shown is the Mconfirm 128-bit random number that contains TK, Pairing Request command, Pairing Response command, initiating device address, and the responding device address. Selecting Frame# 39,600 would provide the Sconfirm random number from the responder (Side 2) with similar information from that device but the random number would be different than Mconfirm.

Once pairing is complete and an encrypted session established, the keys are distributed by the master and slave now identified by Side = M and Side = S respectively in the **Summary** pane. In Frame# 39,661 the slave has distributed LTK to the master to allow exchange of encrypted data. Frame# 39,661 through 39,714 in the Summary pane SMP tab are the key distribution frames.



Figure 24 - SMP Pairing Confirm (Frame# 39,591) from Initiator (Side 1)



Figure 25 - SMP Key Distribution Frames

A.4.7.2.2 Link Layer

The Link Layer (LL) protocol manages the *Bluetooth* low energy radio transmissions and is involved in starting link encryption. To observe the decoded LL commands, click on the **Frame Display LE LL** tab, search for and select ControlPkt "LL_ENC_REQ". This command should originate with Side 1, the initiator of the encryption link. In Figure 11 Frame# 39,617 is selected in the Summary pane and we see the decoded LE LL frame is display in the **Decoder** pane. Shown in this frame packet is the SKDm that is the Master Session Key Diversifier (SKDmaster). In Frame# 39,623 you will find SKDslave that is combined with SKDmaster to create the Session Key (SK). Both SDKs were created using the LTK. Frame# 39,635 through 39,649 in the **LE LL** tab completes starting of the encryption process. After the slave sends LL_START_ENC_RSP (Frame# 36,649) the *Bluetooth* devices can exchange encrypted data, and the ComProbe sniffing device can also receive and decrypt the encrypted data because the appropriate "key" is provided in the **BPA 600 Datasource** window.

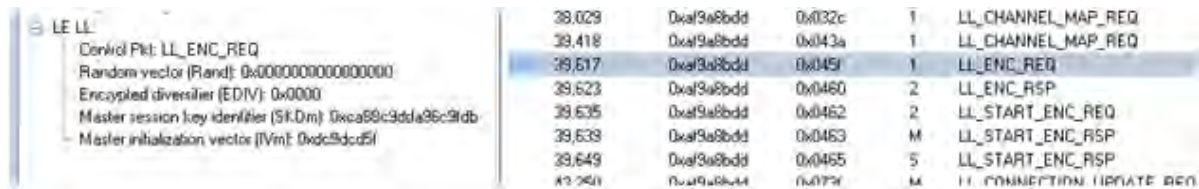


Figure 26 - LE LL Tab Encryption Request (Frame# 39,617) from Initiator (Side 1)

A.4.7.3 Viewing Encryption in the Message Sequence Chart

The ComProbe software **Message Sequence Chart (MSC)** links directly to frames being viewed in the Frame Display. Similarly MSC will display the same information as the **Frame Display Decoder** pane. Frames are synchronized between the **Frame Display Summary** pane and the **MSC**, so clicking on a frame in either window will select that same frame in the other window. Also the protocol tabs are the same in each window. To see the pairing process, click on the SMP tab.

In the image above we see Frame# 35,539 initiating the pairing from the master device. The response, SMP_Pairing Response, is sent from the slave in Frame# 35,545. SMP_Pairing Confirm occurs

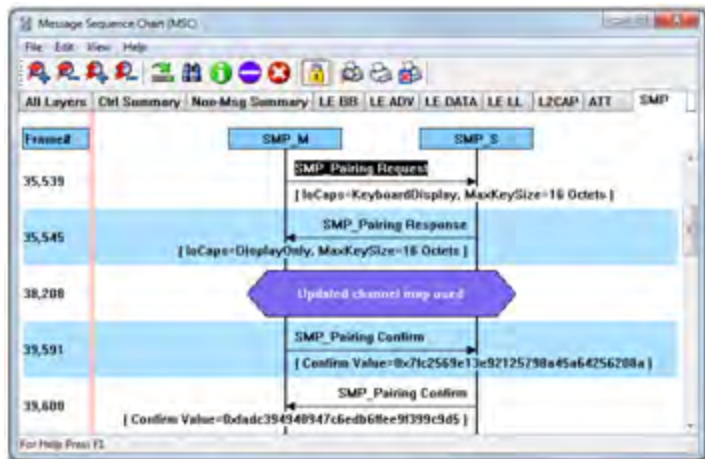


Figure 27 - MSC SMP Pairing (BPA 600 low energy capture)

between the master and the slave devices at Frame# 39,591 and 39,600 respectively.

Clicking on the **MSC** LE LL tab will show the process of encrypting a session link. Clicking on Frame# 39,617 displays the LL_ENC_REQ command from the master to the slave. In the **MSC** below this command you will see the data transferred that includes SKD_{master} used to generate the LTK. At Frame# 39,623 the slave responds with LL_ENC_RSP sending SKD_{slave} to generate LTK at the master. Up to this point all transmissions are unencrypted. For this example the slave sends the request to start encryption, LL_START_ENC_REQ, at Frame#39,635. The master responds with LL_START_ENC_RSP at Frame# 39,639, and finally the slave responds with LL_START_ENC_RSP at Frame# 36,649. At this point the session link is encrypted.

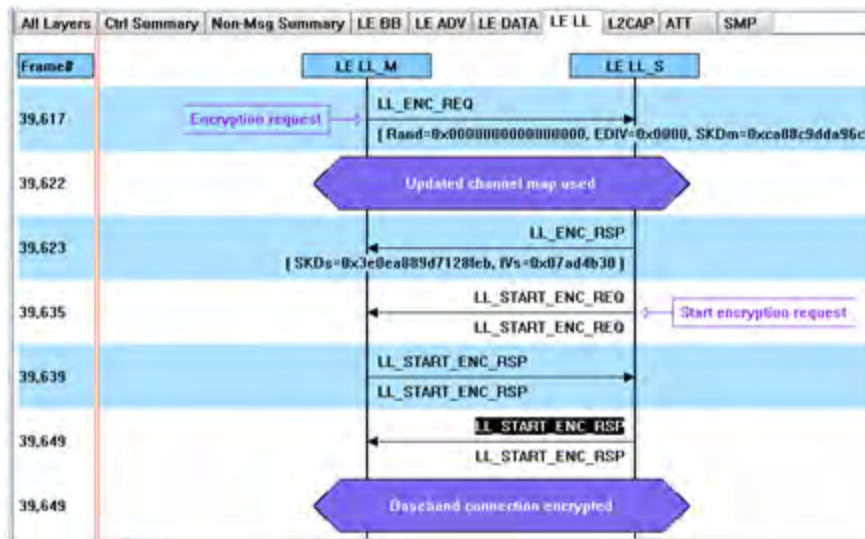


Figure 28 - MSC link Layer Encryption (BPA 600 low energy capture)

A.4.7.4 Viewing Decrypted Data

In the ComProbe software **Frame Display** click on the **LE BB** tab. Search in the **Summary** pane for Decryption Initiated = Yes frames. In the example depicted in the following figure, Frame# 39723 is selected. In the **Decoder** pane LE BB shows that the decryption was initiated and decryption was successful. In LE Data we see the Encrypted MIC value. The MIC value is used to authenticate the sender of the data packet to ensure that the data was sent by a peer device in the link and not by a third party attacker. The actual decrypted data appears between the Payload Length and the MIC in the packet. This is shown in the **Binary** pane below the **Summary** pane.

A.5 Bluetooth® low energy Security

"Paris is quiet and the good citizens are content." Upon seizing power in 1799 Napoleon sent this message on Claude Chappe's optical telegraph. Chappe had invented a means of sending messages line-of-sight. The stations were placed approximately six miles apart and each station had a signaling device made of paddles on the ends of a rotating "regulator" arm whose positions represented code numbers. Each station was also outfitted with two telescopes for viewing the other stations in the link, and clocks were used to synchronize the stations. By 1803 a communications network extended from Paris across the countryside and into Belgium and Italy.

Chappe developed several coding schemes through the next few years. The station operators only knew the codes, not what characters they represented. Not only was Chappe's telegraph system the first working network with protocols, synchronization of serial transmissions but it also used data encryption. Although cryptography has been around for millenniums—dating back to 2000 B.C.—Chappe, was the first to use it in a wide area network in the modern sense.



Figure 30 - Chappe's Optical Telegraph

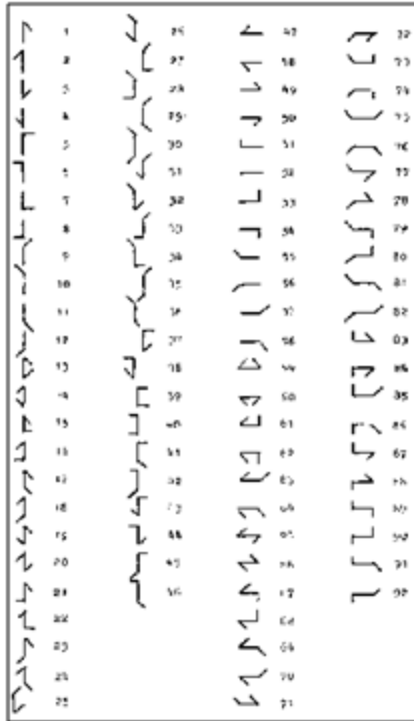


Figure 31 - Chappe's Telegraph Code

Of course anyone positioned between the telegraph stations that had Chappe's telegraph code in hand could decode the transmission. So securing the code was of paramount importance in Chappe's protocol.

Modern wireless networks such as *Bluetooth* low energy employ security measures to prevent similar potentially man-in-the-middle attacks that may have malicious intent.

Bluetooth low energy devices connected in a link can pass sensitive data by setting up a secure encrypted link. The process is similar to but not identical to *Bluetooth* BR/EDR Secure Simple Pairing. One difference is that in *Bluetooth* low energy the confidential payload includes a Message Identification Code (MIC) that is encrypted with the data. In *Bluetooth* BR/EDR only the data is encrypted. Also in *Bluetooth* low energy the secure link is more vulnerable to passive eavesdropping, however because of the short transmission periods this vulnerability is considered a low risk. The similarity to BR/EDR occurs with "shared secret key", a fundamental building block of modern wireless network security.

This paper describes the process of establishing a *Bluetooth* low energy secure link.

A.5.1 How Encryption Works in *Bluetooth* low energy

Data encryption is used to prevent passive and active—man-in-the-middle (MITM) — eavesdropping attacks on a *Bluetooth* low energy link. Encryption is the means to make the data unintelligible to all but the *Bluetooth* master and slave devices forming a link. Eavesdropping attacks are directed on the over-the-air transmissions between the *Bluetooth* low energy devices, so data encryption is accomplished prior to transmission using a shared, secret key.

A.5.2 Pairing

A *Bluetooth* low energy device that wants to share secure data with another device must first pair with that device. The Security Manager Protocol (SMP) carries out the pairing in three phases.

1. The two connected *Bluetooth* low energy devices announce their input and output capabilities and from that information determine a suitable method for phase 2.
2. The purpose of this phase is to generate the Short Term Key (STK) used in the third phase to secure key distribution. The devices agree on a Temporary Key (TK) that along with some random numbers creates the STK.
3. In this phase each device may distribute to the other device up to three keys:
 - a. the Long Term Key (LTK) used for Link Layer encryption and authentication,
 - b. the Connection Signature Resolving Key (CSRK) used for data signing at the ATT layer, and

- c. the Identity Resolving Key (IRK) used to generate a private address.

Of primary interest in this paper is the LTK. CSRK and IRK are covered briefly at the end.

Bluetooth low energy uses the same pairing process as Classic *Bluetooth*: Secure Simple Pairing (SSP). During SSP initially each device determines its capability for input and output (IO). The input can be None, Yes/No, or Keyboard with Keyboard having the ability to input a number. The output can be either None or Display with Display having the ability to display a 6-digit number. For each device in a pairing link the IO capability determines their ability to create encryption shared secret keys.

The Pairing Request message is transmitted from the initiator containing the IO capabilities, authentication data availability, authentication requirements, key size requirements, and other data. A Pairing Response message is transmitted from the responder and contains much of the same information as the initiators Pairing Request message thus confirming that a pairing is successfully negotiated.

In the sample SMP decode, in the figure at the right, note the “keys” identified. Creating a shared, secret key is an evolutionary process that involves several intermediary keys. The resulting keys include,

1. IRK: 128-bit key used to generate and resolve random address.
2. CSRK: 128-bit key used to sign data and verify signatures on the receiving device.
3. LTK: 128-bit key used to generate the session key for an encrypted connection.
4. Encrypted Diversifier (EDIV): 16-bit stored value used to identify the LTK. A new EDIV is generated each time a new LTK is distributed.
5. Random Number (RAND): 64-bit stored value used to identify the LTK. A new RAND is generated each time a unique LTK is distributed.

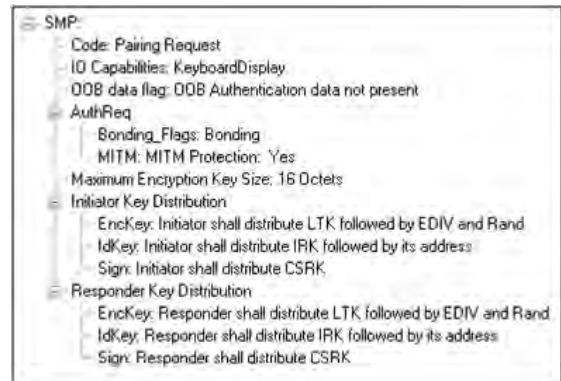


Figure 32 - Sample Initiator Pairing Request Decode (ComProbe Frame Display, BPA 600 low energy capture)

Of particular importance to decrypting the encrypted data on a *Bluetooth* low energy link is LTK, EDIV, and RAND.

A.5.3 Pairing Methods

The two devices in the link use the IO capabilities from Pairing Request and Pairing Response packet data to determine which of two pairing methods to use for generation of the Temporary Key (TK). The two methods are **Just Works** and **Passkey Entry**¹. An example of when **Just Works** method is appropriate is when the IO capability input = None and output = None. An example of when **Passkey Entry** would be appropriate would be if input= Keyboard and output = Display. There are 25 combinations that result in 13 **Just Works** methods and 12 **Passkey Entry** methods.

In **Just Works** the TK = 0. In the **Passkey Entry** method,

$$TK = \begin{cases} 6 \text{ numeric digits, Input} = \text{Keyboard} \\ 6 \text{ random digits, Input} = \text{Display} \end{cases}$$

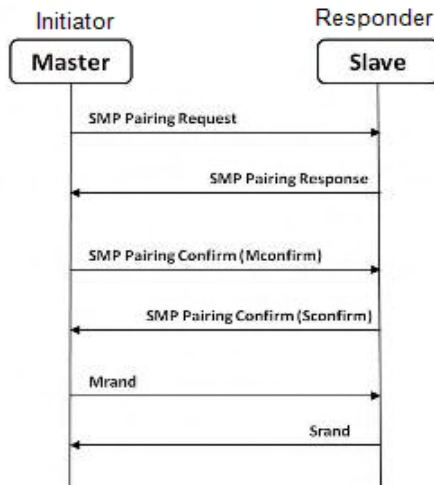
¹A third method, Out Of Band (OOB), performs the same as **Pass Key**, but through another external link such as NFC.



Figure 33 - Initiator Pairing Confirm Example (ComProbe Frame Display, BPA 600 low energy capture)



Figure 34 - Responder Pairing Confirm Example (ComProbe Frame Display, BPA 600 low energy capture)



The initiating device will generate a 128-bit random number that is combined with TK, the Pairing Request command, the Pairing Response command, the initiating device address and address type, and the responding device address and address type. The resulting value is a random number **Mconfirm** that is sent to the responding device by the Pairing Confirm command. The responding device will validate the responding device data in the Pairing Confirm command and if it is correct will generate a **Sconfirm** value using the same methods as used to generate **Mconfirm** only with different 128-bit random number and TK. The responding device will send a Pairing Confirm command to the initiator and if accepted the authentication process is complete. The random number in the **Mconfirm** and **Sconfirm** data is **Mrand** and **Srand** respectively. **Mrand** and **Srand** have a key role in setting encrypting the link.

Finally the master and slave devices exchange **Mrand** and **Srand** so that the slave can calculate and verify Mconfirm and the master can likewise calculate and verify Sconfirm.

Figure 35 - Message Sequence Chart: SMP Pairing

A.5.4 Encrypting the Link

The Short Term Key (STK) is used for encrypting the link the first time the two devices pair. STK remains in each device on the link and is not transmitted between devices. STK is formed by combining **Mrand** and **Srand** which were formed using device information and TKs exchanged with Pairing Confirmation (**Pairing Confirm**).

A.5.5 Encryption Key Generation and Distribution

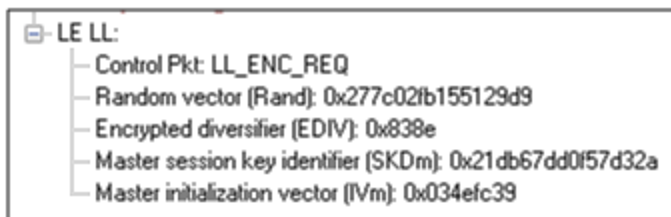


Figure 36 - Encryption Request from Master, Example (ComProbe Frame Display, BPA 600 low energy capture)

To distribute the LTK, EDIV, and Rand values an encrypted session needs to be set up. The initiator will use STK to enable encryption on the link. Once an encrypted link is set up, the LTK is distributed. LTK is a 128-bit random number that the slave device will generate along with EDIV and Rand. Both the master and slave devices can distribute these numbers, but *Bluetooth* low energy is designed to conserve energy, so the slave device is often resource constrained and

does not have the database storage resources for holding LTKs. Therefore the slave will distribute LTK, EDIV, and Rand to the master device for storage. When a slave begins a new encrypted session with a previously linked master device, it will request distribution of EDIV and Rand and will regenerate LTK.

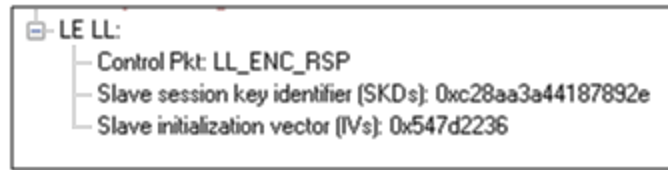


Figure 37 - Encryption Response from Slave, Example
(ComProbe Frame Display, BPA 600 low energy capture)

A.5.6 Encrypting The Data Transmission

Data encryption begins with encrypting the link. The Session Key (SK) is created using a session key diversifier (SKD). The first step in creating a SK is for the master device to send Link Layer encryption request message (LL_ENC_REQ) that contains the SKD_{master}. The SKD_{master} is generated using the LTK. The slave receives SKD_{master}, generates SKD_{slave}, and generates SK by concatenating parts of SKD_{master} and SKD_{slave}. The slave device responds with an encryption response message (LL_ENC_RSP) that contains SKD_{slave}; the master will create the same SK.

Now that a SK has been calculated, the master and slave devices will now begin a handshake process. The slave will transmit unencrypted LL_START_ENC_REQ, but sets the slave to receive encrypted data using the recently calculated SK. The master responds with encrypted LL_START_ENC_RSP that uses the same SK just calculated and setting the master to receive encrypted data. Once the slave receives the master's encrypted LL_START_ENC_RSP message and responds with an encrypted LL_START_ENC_RSP message the *Bluetooth* low energy devices can now begin transmitting and receiving encrypted data.

A.5.7 IRK and CSRK Revisited

Earlier in this paper it was stated that LTK would be the focus, however the IRK and CSRK were mentioned. We revisit these keys because they are used in situations that require a lesser level of security. First let us note that IRK and CSRK are passed in an encrypted link along with LTK and EDIV.

Use of the IRK and CSRK attempt to place an identity on devices operating in a piconet. The probability that two devices will have the same IRK and generate the same random number is low, but not absolute.

IRK and *Bluetooth* low energy Privacy Feature

Bluetooth low energy has a feature that reduces the ability of an attacker to track a device over a long period by frequently and randomly changing an advertising device's address. This is the privacy feature. This feature is not used in the discovery mode and procedures but is used in the connection mode and procedures.

If the advertising device was previously discovered and has returned to an advertising state, the device must be identifiable by trusted devices in future connections without going through discovery procedure again. The IRK stored in the trusted device will overcome the problem of maintaining privacy while saving discovery computational load and connection time. The advertising devices IRK was passed to the master device during initial bonding. The a master device will use the IRK to identify the advertiser as a trusted device.

CSRK and Signing for Authentication

Bluetooth low energy supports the ability to authenticate data sent over an unencrypted ATT bearer between two devices in a trust relationship. If authenticated pairing has occurred and encryption is not required (security mode 2) data signing is used if CSRK has been exchanged. The sending device attaches a digital signature after the data in

the packet that includes a counter and a message authentication code (MAC). The key used to generate MAC is CSRK. Each peer device in a piconet will have a unique CSRK.

The receiving device will authenticate the message from the trusted sending device using the CSRK exchanged from the sending device. The counter is initialized to zero when the CSRK is generated and is incremented with each message signed with a given CSRK. The combination of the CSRK and counter mitigates replay attacks.

A.5.8 Table of Acronyms

| | |
|----------|---|
| CSRK | Connection Signature Resolving Key |
| EDIV | Encrypted Diversifier |
| IO | Input and output |
| IRK | Identity Resolving Key |
| LTK | Long Term Key |
| Mconfirm | 128-bit confirm value from initiator |
| MIC | Message Integrity Check |
| MITM | Man-in-the-middle |
| Mrand | 128-bit random number used to generate Mconfirm |
| OOB | Out of Band |
| RAND | Random Number |
| Sconfirm | 128-bit confirmation value from the responder |
| SK | Session key |
| SMP | Security Manager Protocol |
| Srand | 128-bit random number used to generate Sconfirm |
| SSP | Secure Simple Pairing |
| STK | Short Term Key |
| TK | Temporary Key |

Author: John Trinkle

Publish Date: 21 May 2014

A.6 Bluetooth Virtual Sniffing

A.6.1 Introduction

The ComProbe software Virtual sniffing function simplifies Bluetooth® development and is easy to use. Frontline’s Virtual sniffing with Live Import provides the developer with an open interface from any application to ComProbe software so that data can be analyzed and processed independent of sniffing hardware. Virtual sniffing can also add value to other *Bluetooth* development tools such as *Bluetooth* stack SDKs (Software Development Kits) and *Bluetooth* chip development kits.

This white paper discusses:

- Why HCI sniffing and Virtual sniffing are useful.
- *Bluetooth* sniffing history.
- What is Virtual sniffing?
- Why Virtual sniffing is convenient and reliable.
- How Virtual sniffing works.
- Virtual sniffing and Bluetooth stack vendors.
- Case studies: Virtual sniffing and Bluetooth mobile phone makers.
- Virtual sniffing and you. • Where to go for more information.

A.6.2 Why HCI Sniffing and Virtual Sniffing are Useful

Because the *Bluetooth* protocol stack is very complex, a *Bluetooth* protocol analyzer is an important part of all *Bluetooth* development environments. The typical *Bluetooth* protocol analyzer “taps” a *Bluetooth* link by capturing data over the air. For many *Bluetooth* developers sniffing the link between a *Bluetooth* Host CPU and a *Bluetooth* Host Controller—also known as HCI-sniffing—is much more useful than air sniffing.

HCI-sniffing provides direct visibility into the commands being sent to a *Bluetooth* chip and the responses to those commands. With air sniffing a software engineer working on the host side of a Bluetooth chip has to infer and often guess at what their software is doing. With HCI-sniffing, the software engineer can see exactly what is going on. HCI-sniffing often results in faster and easier debugging than air sniffing.

ComProbe software's Virtual sniffing feature is a simple and easy way to perform HCI-sniffing. Virtual sniffing is not limited to just HCI-sniffing, but it is the most common use and this white paper will focus on the HCI-sniffing application of Virtual sniffing.

It is also important to understand that ComProbe software is a multi-mode product. ComProbe software does support traditional air sniffing. It also supports serial HCI sniffing (for the H4 (HCI UART), H5 (3-wire UART), and BCSP (BlueCore Serial Protocol) protocols), USB HCI (H2) sniffing, SDIO sniffing, and Virtual sniffing. So with ComProbe software nothing is sacrificed—the product is simply more functional than other Bluetooth protocol analyzers.

A.6.3 Bluetooth Sniffing History

Frontline has a strong appreciation for the importance of HCI sniffing because of the way we got involved with *Bluetooth*. Because of our company history, we are uniquely qualified to offer a multi-mode analyzer that provides many ways to sniff and supports a wide variety of protocols. This brief *Bluetooth* sniffing history should help you understand our approach to *Bluetooth* protocol analysis.

In the early days of *Bluetooth*, there were no commercially available *Bluetooth* protocol analyzers, so developers built their own debug tools and/or used protocol analyzers that weren't built for *Bluetooth*. Many developers built homegrown HCI analyzers—basically hex dumps and crude traces—because they recognized the need for visibility into the HCI interface and because it was too difficult to build air sniffers. Several companies developed air sniffers because they saw a market need and because they realized that they could charge a high price (USD \$25,000 and higher).

Two *Bluetooth* chip companies, Silicon Wave and Broadcom were using Frontline's Serialtest® serial analyzer to capture serial HCI traffic and then they would manually decode the HCI byte stream. This manual decoding was far too much work and so, independently, Silicon Wave and Broadcom each requested that Frontline produce a serial HCI *Bluetooth* analyzer that would have all the features of Serialtest. In response to these requests Frontline developed SerialBlue®—the world's first commercially available serial HCI analyzer.

The response to SerialBlue was very positive. When we asked our *Bluetooth* customers what they wanted next we quickly learned that there was a need for an affordable air sniffer that provided the same quality as SerialBlue. We also learned that the ultimate *Bluetooth* analyzer would be one that sniff air and sniff HCI simultaneously.

As work was progressing on our combination air sniffer and HCI sniffer the functional requirements for *Bluetooth* analyzers were changing. It was no longer good enough just to decode the core *Bluetooth* protocols (LMP, HCI, L2CAP, RFCOMM, and OBEX). Applications were beginning to be built on top of *Bluetooth* and therefore application level protocol decoding was becoming a requirement. For example, people were starting to browse the Internet using *Bluetooth*-enabled phones and PDAs therefore a good *Bluetooth* analyzer would need to support TCP/IP, HTTP, hands-free, A2DP, etc.

For Frontline to support for these higher levels protocols was no problem since they were already in use in other Frontline analyzer products. People have been using Frontline Serialtest serial analyzers and Ethertest™ Ethernet analyzer to troubleshoot TCP/IP and Internet problems for many years.

As we continued to work closely with the *Bluetooth* community we also came across one other requirement: sniffing itself had to be made easier. We took a two-pronged approach to this problem. We simplified air sniffing (and we continue to work on simplifying the process of air sniffing) and we invented Virtual sniffing.

A.6.4 Virtual Sniffing—What is it?

Historically, protocol analyzers have physically tapped the circuit being sniffed. For example, an Ethernet circuit is tapped by plugging into the network. A serial connection is sniffed by passively bridging the serial link. A *Bluetooth* air sniffer taps the piconet by synchronizing its clock to the clock of the piconet Master.

Not only is there a physical tap in traditional sniffing, but the sniffer must have some knowledge of the physical characteristics of the link being sniffed. For example, a *Bluetooth* air sniffer must know the BD_ADDR of at least one piconet member to allow it perform clock synchronization. A serial sniffer must know the bit rate of the tapped circuit or be physically connected to the clock line of the circuit.

With Virtual sniffing the protocol analyzer itself does not actually tap the link and the protocol analyzer does not require any knowledge of the physical characteristics of the link.

In computer jargon, “virtual” means “not real”. Virtual memory is memory that doesn’t actually exist. Virtual reality is something that looks and feels real, but isn’t real. So we use the term Virtual sniffing, because there is sniffing taking place, but not in the traditional physical sense.

A.6.5 The Convenience and Reliability of Virtual Sniffing

Virtual sniffing is the most convenient and reliable form of sniffing and should be used in preference to all other forms of sniffing whenever practical. Virtual sniffing is convenient because it requires no setup to use except for a very small amount of software engineering (typically between one and four hours) that is done once and then never again. Once support for Virtual sniffing has been built into application or into a development environment none of the traditional sniffing setup work need be done.

This means:

- NO piconet synchronization.
- NO serial connection to tap.
- NO USB connection to tap.

Virtual sniffing is reliable because there is nothing that can fail. With Virtual sniffing all data is always captured.

A.6.6 How Virtual Sniffing Works

ComProbe software Virtual sniffing works using a feature called Live Import. Any application can feed data into ComProbe software using Live Import. A simple API provides four basic functions and a few other more advanced functions. The four basic Live Import functions are:

- Open a connection to ComProbe software.
- Close a connection to ComProbe software.
- Send an entire packet to ComProbe software.
- Send a single byte to ComProbe software.

All applications that send data to ComProbe software via Live Import use the first two functions. Usually only one of the two Send functions is used by a particular application. When ComProbe software receives data from the application via Live Import, the data is treated just as if it had been captured on a Frontline ComProbe sniffer. The entire protocol stack is fully decoded.

With Virtual sniffing the data can literally be coming from anywhere. ComProbe software does not care if the data being analyzed is being captured on the machine where ComProbe software is running or if the data is being captured remotely and passed into ComProbe software over an Internet connection.

A.6.7 Virtual Sniffing and *Bluetooth* Stack Vendors

As the complexity of the *Bluetooth* protocol stack increases *Bluetooth* stack vendors are realizing that their customers require the use of a powerful *Bluetooth* protocol analyzer. Even if the stack vendor’s stack is bug free,

there are interoperability issues that must be dealt with.

The homegrown hex dumps and trace tools from the early days of *Bluetooth* just are not good enough anymore. And building a good protocol analyzer is not easy. So stack vendors are partnering with Frontline. This permits the stack vendors to concentrate on improving their stack.

The typical *Bluetooth* stack vendor provides a Windows-based SDK. The stack vendor interfaces their SDK to ComProbe software by adding a very small amount of code to the SDK, somewhere in the transport area, right about in the same place that HCI data is sent to the Host Controller.

If ComProbe software is installed on the PC and the Virtual sniffer is running then the data will be captured and decoded by ComProbe software, in real-time. If ComProbe software is not installed or the Virtual sniffer is not running then no harm is done. Virtual sniffing is totally passive and has no impact on the behavior of the SDK.

One Frontline stack vendor partner feels so strongly about ComProbe software that not only have they built Virtual sniffing support in their SDK, but they have made ComProbe software an integral part of their product offering. They are actively encouraging all customers on a worldwide basis to adopt ComProbe software as their protocol analysis solution.

A.6.8 Case Studies: Virtual Sniffing and *Bluetooth* Mobile Phone Makers

Case Study # 1

A *Bluetooth* mobile phone maker had been using a homemade HCI trace tool to debug the link between the Host CPU in the phone the *Bluetooth* chip. They also were using an air sniffer. They replaced their entire sniffing setup by moving to ComProbe software.

In the original test setup the Host CPU in the phone would send debug messages and HCI data over a serial link. A program running on a PC logged the output from the Host CPU. To implement the new system using Virtual sniffing, a small change was made to the PC logging program and it now sends the data to ComProbe software using the Live Import API. The HCI traffic is fully decoded and the debug messages are decoded as well.

The decoder for the debug messages was written using ComProbe software's DecoderScript feature. DecoderScript allows ComProbe software user to write custom decodes and to modify decodes supplied with ComProbe software. DecoderScript is supplied as a standard part of ComProbe software. In this case, the customer also created a custom decoder for HCI Vendor Extensions.

The air sniffer that was formerly used has been replaced by the standard ComProbe software air sniffer.

Case Study # 2

A second *Bluetooth* mobile phone maker plans to use Virtual sniffing in conjunction with a Linux-based custom test platform they have developed. Currently they capture serial HCI traffic on their Linux system and use a set of homegrown utilities to decode the captured data.

They plan to send the captured serial HCI traffic out of the Linux system using TCP/IP over Ethernet. Over on the PC running ComProbe software they will use a simple TCP/IP listening program to bring the data into the PC and this program will hand the data off to ComProbe software using the Live Import API.

A.6.9 Virtual Sniffing and You

If you are a *Bluetooth* stack vendor, a *Bluetooth* chip maker, or a maker of any other products where integrating your product with ComProbe software's Virtual sniffing is of interest please contact Frontline to discuss your requirements. There are numerous approaches that we can use to structure a partnership program with you. We believe that a partnership with Frontline is an easy and cost-effective way for you to add value to your product offering.

If you are end customer and you want to take advantage of Virtual sniffing, all you need to do is buy any Frontline *Bluetooth* product. Virtually sniffing comes standard with product.

Author: Eric Kaplan

Publish Date: May 2003

Revised: December 2013

Index

A

A2DP Decoder Parameters 48

Aborted Frame 308

About Display Filters 110

About L2CAP Decoder Parameters 53

Absolute Time 313

Adaptive Frequency Hopping

- PER Stats 208

Add a New or Save an Existing Template 47

Adding a New Predefined Stack 85

Adding Comments To A Capture File 295

Advanced System Options 307

Apply Capture Filters 112

Apply Display Filters 110-115

ASCII 266

- character set 319
- viewing data in 266

ASCII Codes 319

ASCII Pane 107

Audio Expert System 214

- bitrate 236, 242
- calibratioin 222
- event type

 - Audio 228

 - Clipping 232
 - Dropout 233
 - Glitch 233

 - Bluetooth 225
 - Codec 226
 - frame synchronization 250

- operating mode

 - referenced 217, 222
 - test file 218

- Wave Panel 237

 - viewer 240

- Auto-Sizing Column Widths 104
- Automatically Request Missing Decoding Information 87
- Automatically Restart 305
- Automatically Restart Capturing After 'Clear Capture Buffer' 305
- Automatically Save Imported Capture Files 305
- Autotraversal 85, 87
- AVDTP 48, 50-51
- AVDTP Override Decode Information 51
- Average Throughput Indicators

 - Average Throughput - Selected 137
 - Average_Throughput_Indicators 136

B

Baudot 266, 304

Baudot Codes 319

Begin Sync Character Strip 268

Binary 265, 277

Binary Pane 107

BL 320

Bluetooth Timeline 126

- Audio Expert System 250

Bookmarks 289-290

Boolean 113, 118

BPA 600 7-8, 19, 22-23, 26, 28, 32, 35-36

Broken Frame 267

BS 320



BT Snoop File Format 315

BT Timeline Legend 141

Btsnoop 315

Buffer 294, 305

 Buffer Overflow 305

 Buffer/File Options 305

Byte 108, 264-265, 318

 Searching 280

byte export 99

C

Calculating Data Rates and Delta Times 264

Capture Buffer 294, 305, 307

 Capture Buffer Size 305

Capture File 76, 294-297, 305, 307

 auto-save imported files 305

 capture to a series of files 305

 capture to one file 305

 changing default location of 309

 changing max size of 305, 307

 framing captured data 86

 importing 297

 loading 296

 reframing 86

 removing framing markers 86

 saving 294-295

 starting capture to file 76

Capturing 76

 Data to Disk 76

CFA file 295-296

Changing Default File Locations 309

Character 277, 320

 Character Pane 107

Character Set 266, 319

Choosing a Data Capture Method 5

Clear Capture Buffer 305

CN 320

Coexistence View 161

 Audio Expert System 250

 le Devices Radio Buttons 181

 Legend 182

 Set Button 180

 Throughput Graph 173

 Discontinuities 174

 Dots 176

 Swap Button 175

 Viewport 175

 Zoom Cursor 179

 Zoomed 177

 Freeze Y 178

 Unfreeze Y 178

 Y Scales Frozen 178

 Throughput Indicators 170

 Throughput Radio Buttons 181

 Timeline Radio Buttons 181

 Timelines 182

 discontinuities 190

 high-speed 191

 packet 183

 two timelines 187

Toolbar 168



- Tooltip 174
 - relocate 174, 185
- Color of Data Bytes 108
- Colors 109
- Comma Separated File 301
- Compound Display Filters 113
- Confirm CFA Changes 296
- Context For Decoding 87
- Control Characters 320
- Control Signals 267, 311
- Control Window 17, 305
 - Configuration Information 11
- Conversation Filters 114
- CPAS Control Window Toolbar 10
- CR 320
- CRC 263
- CSV Files 301
- Custom Protocol Stack 84-85
- Custom Stack 84-85
- Customizing Fields in the Summary Pane 104

D

- D/1 320
- D/2 320
- D/3 320
- D/4 320
- D/E 320
- Data 264, 293-294
 - Capturing 76
- Data Byte Color Denotation 108
- Data Errors 285
- Data Extraction 269

- Data Rates 264
- Debug Mode 28, 35
- Decimal 265
- Decode Pane 106
- decoder 321
- Decoder Parameters 45
- DecoderScript 321
- Decodes 44, 84, 88, 95, 106, 274
- decrypt 104
- decryption status 104
- Default File Locations 309
- Delete a Template 48
- Deleting Display Filters 115
- Delta Times 264
- Device Database 36
- Directed Classic Connection 42
- Direction 115
- Directories 309
- Disabling 305
- Discontinuities 141
- Display Filters 110, 115-118
- Display Options 314
- DL 320
- Dots 105
- Duplicate View 98, 262-263
- DUT 26, 28, 32, 35

E

- E/B 320
- E/C 320
- Easy Protocol Filtering 125



EBCDIC 266
 EBCDIC Codes 319

EIR 83

EM 320

EQ 320

Errors 109, 126, 285, 311

ET 320

Event Display 98, 261, 301
 Event Display Export 301
 Event Display Toolbar 262
 Event Numbering 318
 Event Pane 108
 Event Symbols 267

EX 320

Exclude 112

Exclude Radio Buttons 112

Expand All/Collapse All 106

Expand Decode Pane 99

Expert System 214, 251
 event 245

Export
 Export Baudot 304
 Export Events 302
 Export Filter Out 304

Export Payload Throughput Over Time 139

Extended Inquiry Response 83

F

F/F 320

FCSs 263

Field Width 104

File 293-296, 305

File Locations 309

File Series 305

File Types Supported 296

Filtering 124

Filters 110-118, 125

Find 274, 277-278, 280-281, 285

Find - Bookmarks 287

Find Introduction 273

Font Size 269

Frame Display 88, 91, 95, 97-98, 104-109
 Audio Expert System 250
 Frame Display - Change Text Highlight Color 108
 Frame Display - Find 95
 Frame Display Status Bar 95
 Frame Display Toolbar 91
 Frame Display Window 89
 Frame Recognizer Change 268
 Frame Symbols 105

Frame Information on the Control Window 12

Freeze 264

FS 321

G

Go To 280

Green Dots in Summary Pane 105

GS 320

H

Hex 265

Hexadecimal 107

Hiding Display Filters 115

Hiding Protocol Layers 95

High Resolution Timestamping 313



HT 321

I

I/O Settings Change 268

Icons in Data on Event Display 267

Importable File Types 297

Importing Capture Files 296

INCLUDE 112

Include/Exclude 112

Information Screen 40

L

L2CAP 53

 L2CAP Override Decode Information 54

Layer Colors 109

LF 321

Link Key 28

 LSB 28, 35, 79

Live Update 264

Logical Byte Display 95

Logical Bytes 95

Long Break 268

low energy Data Encryption/Master and Slave
 Assignment 103

Low Energy Timeline

 Button Bar/Legend 143

 Discontinuities 156

 Legend 148

 Navigating and Selecting Data 157

 Zooming 158

low energy Timeline Introduction 142-143

Low Power 268

M

Main Window 9

Mesh 58, 61

 CSRmesh 58, 61

 Mesh 58, 61

Message Sequence Chart 194

Message Sequence Chart - Find and Go To 200

Message Sequence Chart - Go To 201

Minimizing 17

Missing Bluetooth Clock 141

Missing Decode Information 50, 56

Mixed Channel/Sides 266

Mixed Sides Mode 266

Modem Lead Names 311

Modify Display Filters 117

Multiple Event Displays 263

Multiple Frame Displays 98

N

NK 320

Node Filters 114

Nonprintables 304

Notes 295

NU 320

Number Set 265

Numbers 319

O

Object Throughput Stats File 140

Octal 265

One_Second_Throughput_Indicators 137

Open 263

 Open Capture File 296

Options 305, 307-308, 311



Other Term

- Subterm 16
- Override Decode Information 51, 54, 57
- Overriding Frame Information 87
- Overrun Errors 286

P

- Packet Error Rate (PER Stats) 204
 - Packet Error Rate 204
 - PER Stats Scroll Bar 212
- Packet Timeline 131, 141
- Packet Timeline Menu Bar 132
- Packet_Depiction 127
- Packet_Navigation_and_Selection 131
- Packet_Timeline_Introduction 126
- Packet_Timeline_Visual_Elements 134
- Panes 98
- Pattern 276
- Pause 76
- Performance Notes 314
- Physical Errors 109
- Printing 300
- Printing from the Frame Display 297
- ProbeSync 4, 8, 79
- Progress Bars 318
- Protocol
 - Protocol Layer Colors 109
 - Protocol Layer Filtering 124
- Protocol Stack 84-85, 87

Q

- Quick Filtering 124, 126

R

- Radix 107, 265
- Red Frame Numbers 109
- Reframe 86
- Reframing 86
- Relative Time 278, 313
- Remove
 - Bookmarks 289-290
 - Columns 105
 - Custom Stack 84
 - Filters 115
 - Framing Markers 86
- Reset Panes 98
- Resolution 312
- Resumed 267
- Revealing Protocol Layers 95
- RFCOMM 55-57
- RFCOMM Missing Decode Information 56
- RFCOMM Override Decode Information 57
- roleless 20, 78
- RS 320
- RSSI 126

S

- Save 112, 293-295
- Save As 293
- Saving 294-295
 - Display Filter 111
 - Imported Capture Files 305
- Saving the Capture File using File > Save or the Save icon 293
- Search 274, 276, 278, 280-281, 285, 288-290
 - binary value 276



- bookmarks 290
- character string 276
- errors 285
- event number 281
- frame number 280
- hex pattern 276
- pattern 276
- special event 281
- timestamp 278
- wildcards 276
- Secure Simple Pairing 28, 35
- Seed Value 263
- Short Break 268
- Side Names 311
- Sides 311
- Signal Strength 126
- Sorting Frames 95
- Special Events 281
- Start 267
- Start Up Options 308
- Summary 101
- Summary Pane 101, 104-105
- Sync Dropped 268
- Sync Found 268
- Sync Hunt Entered 268
- Sync Lost 268
- Synchronization 97
- System Settings 305, 307

T

- Technical Support 323
- Test Device Began Responding 268

- Test Device Stopped Responding 268
- Throughput Displays
 - Throughput_Displays 136
- Throughput Graph 139
- Timestamp 289, 312-313
- Timestamping 289, 311, 313
- Timestamping Disabled 268
- Timestamping Enabled 268
- Timestamping Options 305, 311
- Timestamping Resolution 312
- Timestamps 311, 313
- Transferring Packets 76
- Truncated Frame 268

U

- unable to decrypt 104
- Underrun Error 268
- Unframe 86
- Unframe Function 86
- Unframing 86
- Unknown Event 268

V

- vendor specific decoder 321
- Viewing Data Events 265

W

- Wrap Buffer/File 305

Z

- Zooming 189
 - Zooming 136
- zooming cursor 179

