

TCP & HL7 Streaming Interface



Document Revision: 1.5

Document Date: 30.Oct.2023

Purpose

As of the release of **v1.0.46** (and newer), the **BEDSIDE**, **SERVICE** and **CENTRAL** versions of MediCollector software products will allow users to stream live data out via TCP/IP using one of three protocols:

- **HL7 2.6:** The Health Level Seven standard (HL7) enables MediCollector to stream data to a hospital information systems (HIS), electronic medical records systems (EMR), or integration engine, etc. MediCollector uses HL7 version 2.6. The HL7 interface is explained in more detail on PAGE 6 of this document.
- **HL7 FHIR:** The HL7 Fast Healthcare Interoperability Resources (FHIR) standard is a modern and universal method for transmitting healthcare data in a structured and readable format. Using this standard you can send live data (both numeric and waveform) to external systems as bundles of resources using the RESTful API, as explain on PAGE 14 of this document.
- **MediCollector Format:** This simple packet structure enables MediCollector to quickly and easily stream data to external applications such Matlab or LabVIEW (example code included). The MediCollector Format is explained in more detail on PAGE 4 of this document.

This document is intended for developers/integrators who want MediCollector to send live data to an external application or system. Using this interface, you can send live signals from one of our supported medical devices directly into your own application or system.

Please note the following:

- This interface is intended for advanced programmers or interface engineers. MediCollector does not provide technical support for your own programming!
- Live data will reach the TCP recipient after a short time delay (e.g. about 1-2 seconds). The data transfer from medical device to MediCollector to the TCP recipient is not (and cannot be) perfectly instantaneous.
- MediCollector will only stream to one recipient at a time (not multiple recipients).
- Data that is sent over TCP/IP is NOT encrypted.
- To enable streaming, do this:
 - For MediCollector BEDSIDE: Launch a new session and in the wizard that appears, you just turn streaming ON. It's easy! Just turn it ON and select the port and launch the session.
 - For MediCollector SERVICE: Turn streaming ON when you configure the service using the "MediCollector Service Manager". This is done by clicking the CONFIGURE button in the "MediCollector Service Manager".

The remaining pages of this document explain how to use the two available protocols (either HL7 or MediCollector Format). If you need help deciding which protocol is right for you, please [contact us](#).

How to stream data out

MediCollector **BEDSIDE**, **SERVICE** and **CENTRAL** can be used to stream live data out. To enable streaming, follow these instructions, depending on which product you are using:

BEDSIDE: Launch a new session by selecting FILE>NEW from the main menu.

SERVICE: Click the CONFIGURE button in the MediCollector Service Manager to configure the service.

The above instructions will open a wizard where you can enable streaming in Step 4 as shown below.

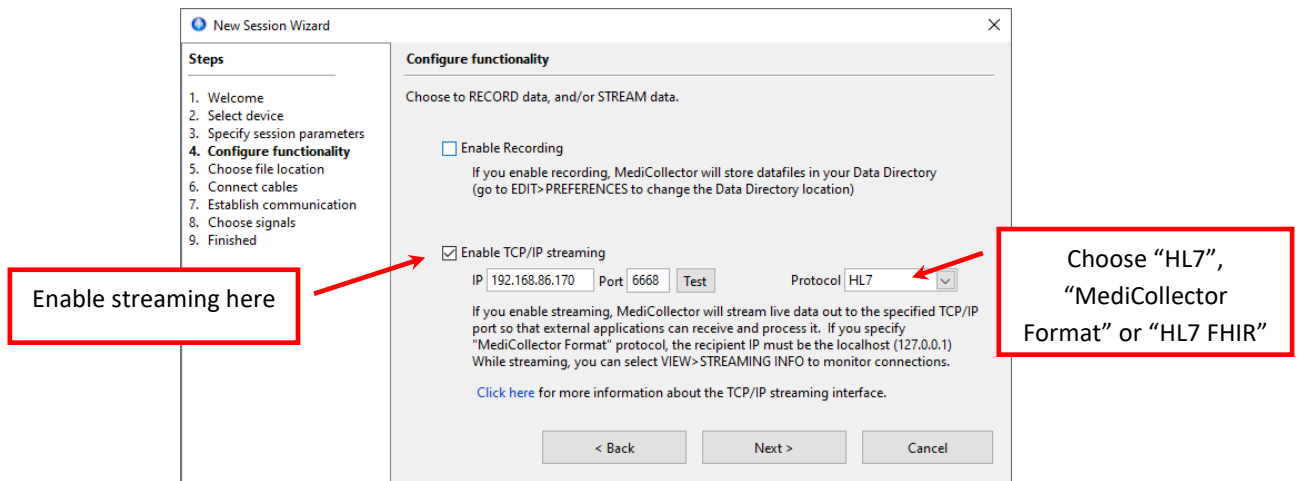
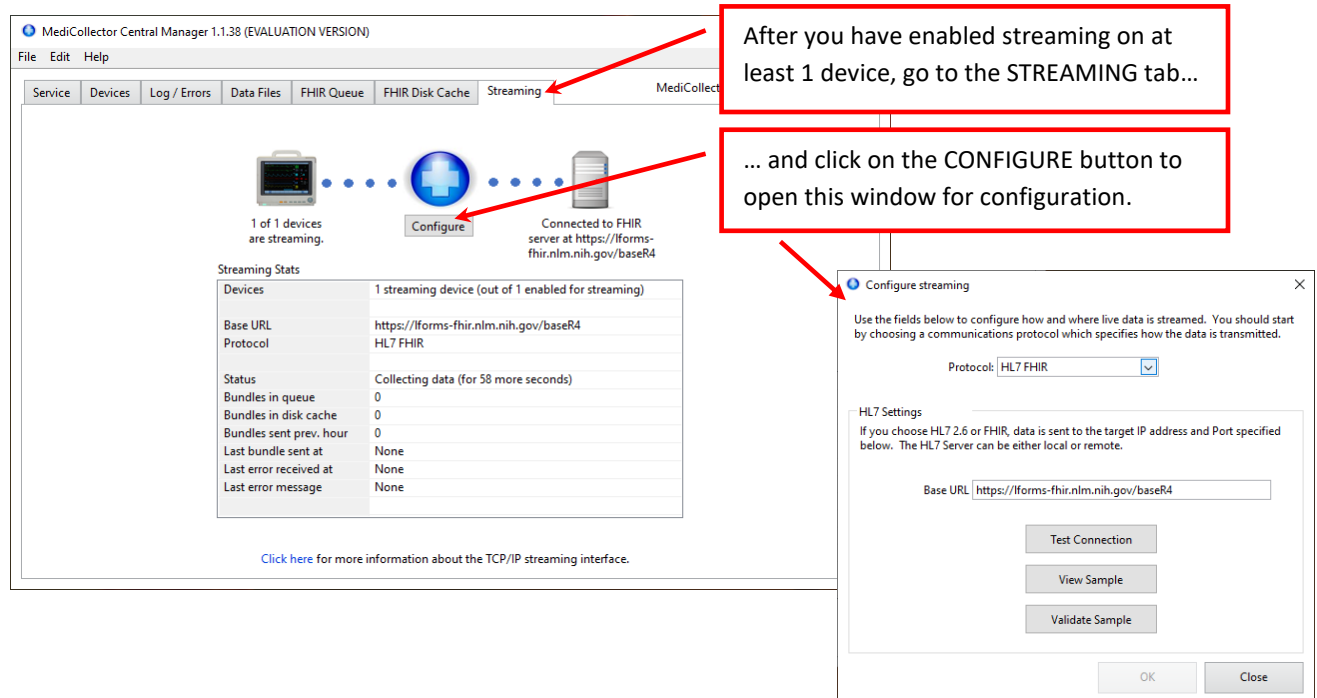


Figure 1 How to enable streaming

CENTRAL: Create a new device (or double-click on an existing device) in the MediCollector Central Manager. This will show a wizard like Figure 1 above where you can enable streaming for the device. However, to configure streaming, you must go to the STREAMING tab of the MediCollector Central Manager as shown below.



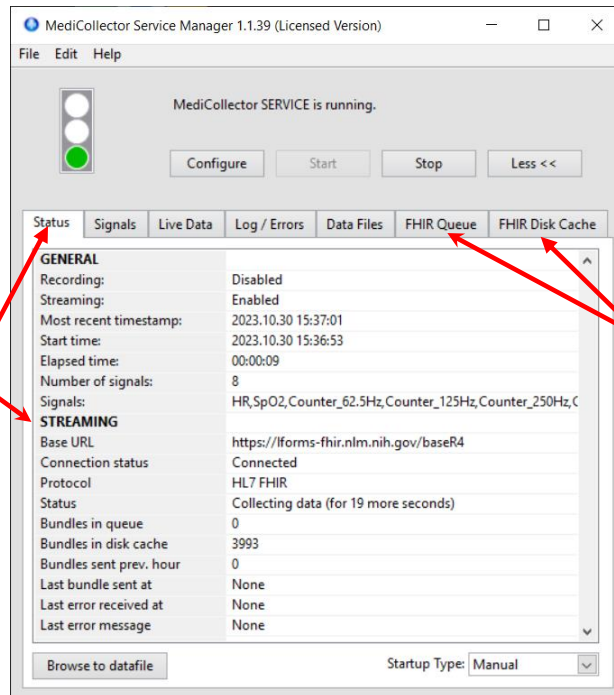
How to monitor the streaming process

Streaming can be monitored by following the instructions below:

If using **MediCollector BEDSIDE**, click on the HL7, TCP or FHIR buttons in the toolbar to open a window for monitoring the streaming process.



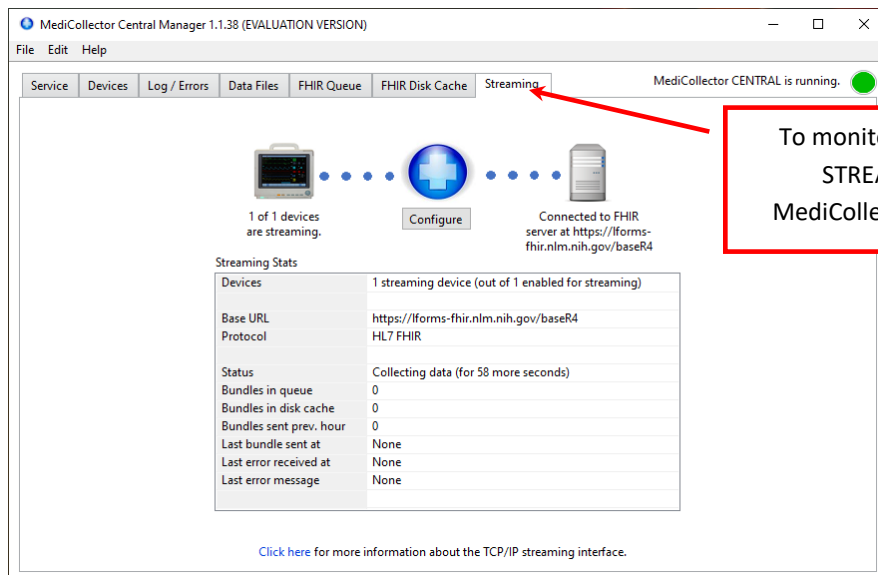
If using **MediCollector SERVICE**, use the MediCollector Service Manager to monitor streaming as shown below:



Streaming info will appear in the STATUS tab

And if using HL7 or FHIR, you will see additional tabs for monitoring streaming

If using **MediCollector CENTRAL**, use the MediCollector Central Manager to monitor streaming as shown below:



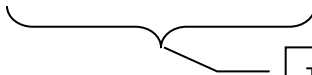
To monitor streaming, use the STREAMING tab in the MediCollector Central Manager

MediCollector Format

This format is intended for programmers who want to send live data quickly and easily to external applications such as Matlab or LabVIEW or your own application. Example programs for receiving and displaying data in Matlab and LabVIEW are provided when you install MediCollector.

Using this format, MediCollector will send data in “packets” as soon as they are received from the medical device. Each packet contains a varying number of data datapoints for a signal (1 or more datapoints per packet). Each packet is formatted as seen below. There are 4 fields in each packet. The first 3 fields have fixed widths as specified below. The last field contains the actual datapoints and has a varying length depending on the number of datapoints.

[Room][Bed][DeviceName][SignalName][SignalFrequency][Tzero][Datapoints][TerminationChar]



These three fields are only present when streaming from the MediCollector CENTRAL product

Field	Field Length	Description
[Room]	30 chars	This field is only present when streaming from the MediCollector CENTRAL product. The room where the device is located, as defined in MediCollector CENTRAL.
[Bed]	30 chars	This field is only present when streaming from the MediCollector CENTRAL product. The bed where the device is located, as defined in MediCollector CENTRAL.
[DeviceName]	30 chars	This field is only present when streaming from the MediCollector CENTRAL product. The name of the device generating the signal, as defined in MediCollector CENTRAL.
[SignalName]	30 chars	The name of the signal contained in this packet, padded with blank spaces to fill the 30 chars width.
[SignalFrequency]	10 chars	The frequency of the signal in Hz, , padded with blank spaces to fill the 10 chars width. This value always uses “.” as the decimal point.
[Tzero]	23 chars	The timestamp of the first datapoint in the packet using the fixed format “YYYY.MM.DD HH:MM:SS.000”. This field is ALWAYS 23 chars in length, always uses “.” as the decimal point, always uses 24 hour time, and always has 3 values after the decimal point.
[Datapoints]	Varies	An array of 8-byte doubles
[TerminationChar]	5 chars	Always equals ‘ ’

To process packets, you should build a program which waits for a termination character (|||||) and then processes each packet according to the specs above. To determine how many datapoints are in each packet (because it will vary), you need to divide the length of the Datapoints field by 8 (because there are 8 bytes in each datapoint).

Example Packets

Below is an example TCP packet sent from MediCollector BEDSIDE or MediCollector SERVICE:

“SlowRandom 0.976562 2017.09.02 14:39:17.863pT£'r¶#@q¹X²~i"@|||||”

The corresponding field values are:

SignalName = SlowRandom

SignalFrequency = 0.976562 Hz

Timestamp = 2017.09.02 14:39:17.863

Datapoints = Two values: 9.85634 and 9.46421

In hex, these 16 bytes are: FE54 A327 72B6 2340 71B9 58B2 ACED 2240

Below is an example TCP packet sent from MediCollector CENTRAL. The CENTRAL version of our product can stream data from multiple devices, so each packet requires three additional fields in order to identify which device generated the data, as demonstrated below:

```
"Room 123A      Bed 789      PM001      SlowRandom      0.976562 2017.09.02
14:39:17.863$TM™™™™™™™™™™@ffffff@|||||
```

The corresponding field values are:

```
Room           =      Room 123A
Bed            =      Bed 789
DeviceName     =      PM001
SignalName     =      SlowRandom
SignalFrequency =      0.976562 Hz
Timestamp      =      2017.09.02 14:39:17.863
Datapoints     =      Two values: 5.4 and 5.6
```

In addition to the physiologic signals, MediCollector sends a status packet (named "MediCollectorStatus") which informs the TCP client of MediCollector's status. This packet is sent (at least) every 5 seconds. When this packet is sent, the value for the "datapoints" field will contain a text string with the value "OK" if MediCollector is acquiring data as it should. If data is not being acquired, this field will contain an error string. Below are some examples of status packets:

```
"MediCollectorStatus 0.2 2020.05.31 17:51:37.000OK"
```

```
"MediCollectorStatus 0.2 2020.05.20 14:39:17.863Error 5015 occurred: A connection error has
occurred. Unable to connect to a patient monitor at the specified COM port."
```

Example Code

Example programs in Matlab and LabVIEW can be found at: **C:\Program Files\MediCollector PRODUCT\Streaming**

HL7 2.6

The Health Level Seven (HL7) standard is a communications protocol used to exchange data between computer systems in a healthcare environment. You can now use this protocol in MediCollector to stream vital signs data to external systems, such as a Healthcare Information system (HIS) or an Integration Engine (such as Mirth or Iguana).

The following pages describe the syntax, structure and flow of HL7 messages sent from MediCollector.

HL7 Rules

1. MediCollector uses HL7 version 2.6.
2. MediCollector sends vital signs using unsolicited observation result messages (message type = ORU^R01).
3. MediCollector sends messages using the LLP (aka MLLP) transport protocol over TCP/IP. MediCollector acts as a TCP/IP client when sending HL7 messages, which means the recipient of your HL7 messages (i.e. your HIS or Integration Engine) must act as the TCP/IP Server. Messages can be sent to the localhost or to a remote server.
4. MediCollector uses numeric codes from the Medical Device Interface Language (MDIL) nomenclature to uniquely identify each vital sign observation. For example, an HR observation would be identified in OBX-3 as 00024182^HR^MDIL.
5. Currently, only “numeric” vital signs are sent via HL7. A “numeric” vital sign is one that is received at 1 second intervals or slower (i.e. the signal frequency ≤ 1 Hz). These are signals such as BP, HR, Temp and SpO2 and are usually displayed on a patient monitor as a simple number on the screen.



Coming Soon! MediCollector does not currently support sending waveforms like ECG, RESP and PLETH via HL7. We are working on this as quickly as possible. Please [contact us](#) if you require waveform support right away.

6. Messages are sent without encryption. It is up to the end-user to encrypt their network (e.g. using VPN tunnels).
7. When sending HL7 messages, MediCollector will attempt to connect to the IP address and port that you have specified. If unable to connect (or if disconnected in the middle of a session), MediCollector will continually try to re-establish a connection every 10 seconds. While disconnected, outgoing HL7 messages will continue to build up in the HL7 Queue. When reconnected, messages in this queue will begin sending again.
8. Datapoints from your medical device are collected into one outgoing HL7 message every 30 seconds.
9. It is important to understand the difference between the **HL7 Queue** and the **HL7 Disk Cache**. When sending data via HL7, MediCollector first puts outbound messages in the HL7 Queue, and then sends them one by one (first in, first out). When a message is successfully sent (e.g. an ACK(AA) is received), it is removed from the HL7 Queue. Under certain circumstances, which are explained below, messages may get removed from the HL7 Queue. When this happens, they are dumped to the HL7 Disk Cache. The following rules explain this in more detail.

10. In addition to the **HL7 Queue** which resides in memory, MediCollector creates an **HL7 Disk Cache** for storing HL7 messages to disk. Messages are written to these folders under various circumstances as explained below. Messages written to the **HL7 Disk Cache** will NOT be re-sent.

📁 **C:\Program Files\MediCollector PRODUCT\HL7**

This is the top-level folder of the **HL7 Disk Cache** containing the various subfolders.

📁 **AE - Application Error**

Messages are written here when the recipient replies with an ACK (AE) response. These messages will NOT be re-sent.

📁 **AR - Application Reject**

Messages are written here when the recipient replies with an ACK (AR) response. These messages will NOT be re-sent.

📁 **No Response**

Messages are written here if the recipient does not reply with a recognizable ACK (after 3 attempts). These messages will NOT be re-sent.

📁 **Unsent**

Messages will be written here if they remain in the queue for over 1 hour. This can occur if, for example, the network connection is lost for over an hour. These messages will NOT be re-sent.

11. Messages that remain in the queue for more than 1 hour will be written to the HL7 Disk Cache and not sent. These old messages will NOT be re-sent. If you want to retrieve these old unsent messages, you will find them in the “Unsent” folder in the HL7 Disk Cache.

12. MediCollector will send 1 message at a time. Datapoints from your medical device are collected into one outgoing HL7 message every 30 seconds. After sending, MediCollector will wait for an acknowledgement (ACK). The next message in the queue will not be sent until the previous message is acknowledged according to the rules below:

Response	Action
ACK (AA)	Recipient accepts message. MediCollector moves on to send the next message.
ACK (AR)	Recipient has rejected the message. MediCollector writes the error to the log and saves the HL7 message to the HL7 Disk Cache in the “AR - Application Reject” subfolder. The message will not be resent. MediCollector then moves on to the next message.
ACK (AE)	Recipient reports an error processing the message. MediCollector writes the error to the log and saves the HL7 message to the HL7 Disk Cache in the “AE - Application Error” subfolder. The message will not be resent. MediCollector then moves on to the next message.
No response	If MediCollector does not receive any response after 10 seconds, the message will be resent. After 3 tries, the messages saves the HL7 message to the HL7 Disk Cache in the “No response” subfolder. The message will not be resent. MediCollector then moves on to the next message.

13. If MediCollector loses its TCP/IP connection to the recipient, all messages will remain in the queue and MediCollector will attempt to re-connect every 10 seconds. While disconnected, your medical device will continue adding more messages to the queue, causing it to grow. As stated earlier in this document, messages that are older than 1 hour will be dumped to the HL7 Disk Cache. By dumping these old messages to the HL7 Disk Cache, this will prevent the queue from growing indefinitely, and will therefore prevent a memory overload.

HL7 Message Packetization

Each HL7 message is sent using LLP (aka MLLP). This means that each message is wrapped using a header and trailer to signify the beginning and end of a message, as seen in Figure 2 below:



Figure 2 HL7 Message Packetization

The header is a vertical tab character (0x0B) which signifies the start of a new message. And the end of a message is marked with the field separator character (0x1C) followed by a carriage return (0x0D).

Nomenclature

MediCollector uses numeric codes from the Medical Device Interface Language (MDIL) nomenclature to uniquely identify each vital sign observation. For example, an HR observation would be identified in OBX-3 as *00024182^HR^MDIL*.

HL7 Message Structure

Each outgoing HL7 message contains the following segments. Each segment is explained in more detail below:

- Message Header (MSH)
- Patient Identification (PID)
- Patient Visit (PV1)
- Observation Request (OBR)
- Observation Result (OBX)
- Notes and Comments (NTE)

Message Header (MSH) Segment

The MSH segment contains message envelope information. The following table describes each field in this segment.

Seq	Field	Description	DT	Len	Sample Value
1	Field Separator	The character that separates the segment ID (MSH) from the first data field (Encoding Characters) in the segment. MediCollector always sets this to a pipe char.	ST	1	
2	Encoding Characters	The four separator characters for the message in this order: component separator (^), repetition separator (~), escape character (\), subcomponent separator (&).	ST	4	^~\&
3	Sending Application	This value can be edited when launching a session. The default value is shown to the right.	HD	227	MediCollector
4	Sending Facility	This value can be edited when launching a session. The default value is shown to the right.	HD	227	SendingFacility
5	Receiving Application	This value can be edited when launching a session. The default value is shown to the right.	HD	227	ReceivingApplication
6	Receiving Facility	This value can be edited when launching a session. The default value is shown to the right.	HD	227	ReceivingFacility
7	Message timestamp	The date and time when the message <i>was created</i> . This is NOT the timestamp when the measurement was taken by the medical device, which is in the OBR segment. Note that time is specified with up to 3 decimal places as shown to the right, in this format: YYYYMMDDHHMMSS.XXX	TS	26	20191107205330.109
9	Message Type	Unsolicited Observation Reporting (ORU) message. Unsolicited transmission of results: R01.	MSG	15	ORU^R01^ORU_R01
10	Message Control ID	The unique identifier for this message. MediCollector uses an elapsed time and a rolling counter to generate unique message IDs.	ST	20	6264756826951_567
11	Processing ID	All messages sent by MediCollector are production messages (P), except for the sample/test messages that can be sent when configuring a session. These sample/test messages are T (Training).	PR	3	P or T
12	Version	The HL7 version used to create this message.	VID	60	2.6
15	Accept Acknowledgement Type	Type of Accept ACK expected. This type of ACK is not required.	ID	2	NE
16	Application Acknowledgement Type	Type of Application ACK expected. MediCollector always requires an application ACK response, so this field is AL.	ID	2	AL
24	Sending Network Address	MediCollector will populate this field with the MAC address of the computer running the MediCollector software.	HD		^DE:AD:BE:EF:00:01^MAC

Patient Identification (PID) Segment

The PID segment identifies the subject of the measurements. The following table describes each field in this segment.



Coming Soon! MediCollector does not currently send patient identifiers in the PID segment. We are working on retrieving patient identifiers from connected medical devices. Please [contact us](#) if you require PID support.

Seq	Field	Description	DT	Len	Sample Value
1	Set ID	Identifies this segment in the message. MediCollector only sends HL7 messages with one PID, so this is always 1.	SI	4	1
3	Patient ID List	MediCollector does not specify patient information yet. For now, this field will be UNKNOWN.	CX	250	UNKNOWN
5	Patient Name	MediCollector does not specify patient information yet. For now, this field will be UNKNOWN.	XPN	250	UNKNOWN

Patient Visit (PV1) Segment

The PV1 segment identifies the patient type and the location where the measurement was taken. The following table describes each field in this segment.

Seq	Field	Description	DT	Len	Sample Value
1	Set ID	Identifies the segment in the message. MediCollector only sends HL7 messages with one PV1 segment, so this is always 1.	SI	4	1
2	Patient Class	This field is used by systems to categorize patients by site. It does not have a consistent industry-wide definition. MediCollector will identify all patients as UNKNOWN (U).	IS	1	U
3	Assigned Patient Location	This field identifies the patient location where the measurement was taken. There are 3 subfields as described below.	PL	80	^Room123^Bed456
3.1	Point of Care	Not implemented	IS	20	Not implemented
3.2	Room	This is the patient room where the measurement was taken. MediCollector will populate this field with the value of the FIRST SESSION PARAMETER specified by the user. When you configure a MediCollector session, you can edit these session parameters to meet your needs (i.e. you can specify the Room Number when you configure and launch your session).	IS	20	Room123
3.3	Bed	This is the patient bed where the measurement was taken. MediCollector will populate this field with the value of the SECOND SESSION PARAMETER. When you launch a MediCollector session, you can edit these session parameters to meet your needs (i.e. you can specify the Bed Number when you configure your session).	IS	20	Bed456

Observation Request (OBR) Segment

The Observation Request (OBR) segment is used to transmit information specific to an order for an observation. There can be multiple OBR segments in an HL7 message. The following table describes each field in this segment.

Seq	Field	Description	DT	Len	Sample Value
1	Set ID	Identifies the segment in the message. The first OBR segment will be 1. Additional OBR segments will increment this ID.	SI	4	1
2	Placer Order Number	Not implemented	EI	22	
3	Filler Order Number	Not implemented	EI	22	
4	Universal Service Identifier	MediCollector uses Medical Device Interface Language (MDIL) to identify observations.	CE	250	MDIL
7	Observation Date/Time	Date and time of the measurement. Note that the time is specified with up to 3 decimal places: YYYYMMDDHHMMSS.XXX	TS	26	20191107205330.109

Observation Result (OBX) Segment

The OBX segment contains the actual measurement values. There can be multiple OBX segments in each OBR segment. The following table describes each field in this segment.

Seq	Field	Description	DT	Len	Sample Value
1	Set ID	Identifies the segment in the message. The first OBX segment will be 1. Additional OBX segments will increment this ID.	SI	4	1
2	Value Type	Format of the measurement. MediCollector fixes this field to NM (numeric).	ID	2	NM
3	Observation Identifier	Contains a unique MDC identifier for the measurement.	CE	250	150456^MDC_PULS_OXIM_S AT_O2^MDC
4	Observation Sub-ID	This field is used to distinguish between multiple observations with the same Observation Identifier. Examples of when this occurs are when the connected medical device has two pulse oximeters (one on left and right hand) or when a pulse is derived from both an ECG and also from a pulse oximeter. When this occurs, the first OBX segment will have OBX-4 = 1 and the second will have OBX-4 = 2.	ST	20	1
5	Observation Value	The value of the vital sign measurement.	varies	varies	99
6	Units	The unit of the vital sign measurement using MDC codes.	CE	250	MDC_DIM_PERCENT
11	Observation Result Status	This field describes the completion status of the measurement. MediCollector only sends the value R = Result Entered, Not Verified. This is because the values are read by a machine and not yet verified by a clinician.	ID	1	R
18	Equipment Instance Identifier	This field identifies the equipment responsible for producing the observation. MediCollector will automatically populate this field with the MAC address of the connected medical device.	EI	22	AA:BB:CC:DD:EE:FF^MAC

Notes and Comments (NTE) Segment

The NTE segment is used to provide descriptive text about each vital sign that is measured. For every OBR segment, a single NTE segment will follow to provide descriptive text about the vital sign. If descriptive text is not available for a given vital sign, this segment will not be included.

Seq	Field	Description	DT	Len	Sample Value
1	Set ID	Identifies the segment in the message. Only one NTE segment will follow each OBR, so this field is always 1.	SI	4	1
2	Source of Comment	This field is set to O (OTHER). The source of this comment is from the connected medical device hardware or from the manufacturer's documentation.	ID	2	O
3	Comment	This field provides descriptive text about the signal. An example is shown to the right.	FT		Peripheral oxygen saturation

Sample HL7 2.6 Message

Below is an example HL7 2.6 message generated by MediCollector.

```
MSH|^~\&|MediCollector|SendingFacility|ReceivingApplication|ReceivingFacility|20200116133
348.000||ORU^R01^ORU_R01|6324968285711_000|T|2.6||NE|AL|||||^44:85:00:A0:06:31^MAC
PID|1||UNKNOWN||UNKNOWN|||
PV1|1|U|^Room123^Bed456||
OBR|1||MDIL|||20200116133338.000|
OBX|1|NM|00024182^HR^MDIL||69|0004-0AA0^bpm^MDIL||||F|||||AA:BB:CC:DD:EE:FF^MAC
NTE|1|O|Heart rate derived from ECG
OBX|2|NM|00024BB8^SpO2^MDIL||99|0004-0220^%^MDIL||||F|||||AA:BB:CC:DD:EE:FF^MAC
NTE|1|O|Peripheral oxygen saturation (pulse oximeter)
OBR|2||MDIL|||20200116133339.000|
OBX|1|NM|00024182^HR^MDIL||70|0004-0AA0^bpm^MDIL||||F|||||AA:BB:CC:DD:EE:FF^MAC
NTE|1|O|Heart rate derived from ECG
OBX|2|NM|00024BB8^SpO2^MDIL||98|0004-0220^%^MDIL||||F|||||AA:BB:CC:DD:EE:FF^MAC
NTE|1|O|Peripheral oxygen saturation (pulse oximeter)
```

Figure 3 Sample HL7 Message

In this example, please note the following:

- The first MAC address in MSH-24 is from the computer running the MediCollector software
- The other MAC addresses, in OBR-18, are retrieved from the connected medical device
- The first timestamp in MSH-7 is when the message was created. This time differs from the other timestamps (in OBR-7), which is when the observation was obtained.

FHIR

If you choose to stream data using the FHIR standard, MediCollector will format your outgoing data into Observation resources and then collect them into bundles and transmit them to your chosen target FHIR Server. Data from your device(s) are buffered in memory and then bundled up and sent once per minute using the RESTful API.

Resources are formatted using a “flat” resource model as shown in the sample resource in Figure 4 (next page). Signals are encoded using MDC and/or LOIC codes. MediCollector will attempt to transmit each bundle up to 3 times. If unable to send a bundle after 3 attempts (e.g. due to loss of network connectivity), the bundle will be saved locally to disk in the FHIR Disk Cache and can then be manually re-sent once the problem has been resolved.

Subject

Currently, our products do not have the ability to identify which patient is connected to your medical device. Therefore, for simplicity’s sake, the subject of each outgoing resource is identified as “Unknown” as shown below.

```
"subject":{"display":"Unknown"}
```



Coming Soon! MediCollector does not currently identify the subject (patient) in outgoing FHIR resources, but we are working on it. Please [contact us](#) if you require patient identification.

FHIR Disk Cache

MediCollector creates the FHIR **Disk Cache** for storing FHIR messages to disk. FHIR bundles are written to these folders under various circumstances as explained below. Messages written to the **FHIR Disk Cache** will NOT be re-sent without user intervention (see below).

C:\Program Files\MediCollector PRODUCT\FHIR

This is the top-level folder of the **FHIR Disk Cache** containing the various subfolders.

Error

Messages are written here if MediCollector receives an error when sending the bundle (after 3 attempts).

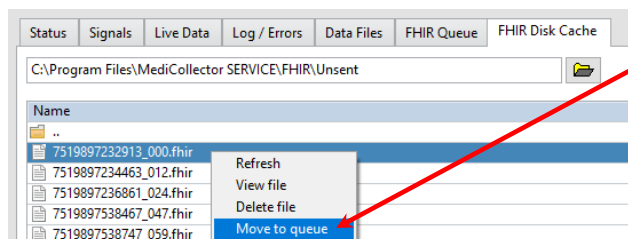
No Response

Messages are written here if the target FHIR Server does not respond to a sent message (after 3 attempts)

Unsent

Messages will be written here if they remain in the queue for over 1 hour. This can occur if, for example, the network connection is lost for over an hour.

MediCollector will not attempt to re-send messages stored in the FHIR Disk Cache unless the user chooses to do so. The user can manually choose to re-send messages by viewing the FHIR Disk Cache and right-clicking on a message and selecting **Move to queue** as shown below.



To attempt to re-send a bundle stored in the HL7 Disk Cache, select the bundle and choose **Move to queue**

Sample HL7 FHIR Message

Below is an example HL7 FHIR message generated by MediCollector.

```
{
  "resourceType": "Observation",
  "meta": { "profile": [ "http://hl7.org/fhir/StructureDefinition/devicemetricobservation" ] },
  "text": { "status": "generated", "div": "<div xmlns='http://www.w3.org/1999/xhtml'>This observation contains multiple values
(valueSampledData). Most recent value is Pulse = 60 .</div>" },
  "status": "final",
  "category": [ { "coding": [ { "system": "http://terminology.hl7.org/CodeSystem/observation-category", "code": "vital-
signs", "display": "Vital Signs" } ] } ],
  "code": { "coding": [ { "system": "urn:iso:std:iso:11073:10101", "code": "2147701761", "display": "Pulse" } ] }, "text": "Pulse",
  "subject": { "display": "Unknown" },
  "effectiveDateTime": "2022-06-25T13:37:02.512-05:00",
  "valueSampledData": {
    "origin": { "value": 0 },
    "period": 1.024,
    "dimensions": 1,
    "data": "60 60 60 60 60 60",
  }
  "performer": [ { "display": "MediCollector" } ],
  "device": { "reference": "#devicemetric1" },
  "contained": [ {
    "resourceType": "DeviceMetric",
    "id": "devicemetric1",
    "meta": { "profile": [ "http://hl7.org/fhir/StructureDefinition/DeviceMetric" ] },
    "text": { "status": "generated", "div": "<div xmlns='http://www.w3.org/1999/xhtml'>Pulse measurement</div>" },
    "type": { "coding": [ { "system": "urn:iso:std:iso:11073:10101", "code": "2147701761", "display": "Pulse" } ] }, "text": "Pulse",
    "source": { "reference": "#device1" },
    "category": "measurement",
    "measurementPeriod": { "repeat": { "frequency": 1, "period": 1.024, "periodUnit": "s" } }
  }, {
    "resourceType": "Device",
    "id": "device1",
    "text": { "status": "generated", "div": "<div xmlns='http://www.w3.org/1999/xhtml'>Philips Intellivue Series Patient Monitor
that generated the observation</div>" },
    "manufacturer": "Philips",
    "identifier": [ { "system": "urn:oid:1.2.840.10004.1.1.1.0.0.1.0.0.1.2680", "value": "0009FB08626F" } ],
    "deviceName": [ { "name": "Patient Monitor", "type": "user-friendly-name" } ],
    "type": { "text": "Philips Intellivue Series" },
    "location": { "display": "Room 5/Bed 9/TABBY" } } ]
}
```

For this proof-of-concept,
the patient is "Unknown"

For simplicity, the Device
and DeviceMetric references
are "contained"

Figure 4. Example FHIR resource