

Automation Interface



For MediCollector BEDSIDE

Document Revision: 5.7

This document explains how to automate MediCollector BEDSIDE. Using this automation interface, you can program a third party application to launch and automatically begin recording or streaming data in MediCollector BEDSIDE and also monitor MediCollector BEDSIDE's status.

Requirements: You must have MediCollector version 1.0.11 or newer in able to use automation.

To enable automation, first SHUTDOWN MEDICOLLECTOR and then cut & paste the following section of text into your *settings.ini* file which you will find in *C:\Program Files\MediCollector BEDSIDE*:

```
[Automation]
Automate = TRUE
EnableRecording = TRUE
EnableStreaming = FALSE
StreamingPort = 2057
StreamingIP="127.0.0.1"
StreamingProtocol="HL7"
DisableLiveGraphs = FALSE
DeviceName = "Philips Intellivue Series"
SignalNames = "HR,Pulse,Resp"
SignalsToDisplay = "HR,Pulse"
COMPort = "COM5"
SessionParameter0 = "MySessionID"
SessionParameter1 = "MySubjectID"
PathToDataFile = ""
SplitFiles = 1
PathToAutomationLog = "C:\Program Files\MediCollector BEDSIDE\automationlog.log"
PathToSignalListFile = "C:\Program Files\MediCollector BEDSIDE\signlistfile.txt"
SecondsBetweenRecordActiveMsgs = 15
```

Each row above is called a "key". Note how these keys are located in the [Automation] section of the INI file. This section can be located anywhere inside the INI file, as long as it is present. You can modify these keys to control how the automation will function. You should not edit any keys within other sections of the INI file.

Below is an explanation of each key

- **Automate = TRUE**
Change this value to TRUE or FALSE in order to turn ON or OFF the automation process. If you launch MediCollector with *Automate=True*, then MediCollector will attempt to automatically begin a new recording using the additional settings described below. If you disable automation by setting *Automate = FALSE*, then all of the other settings/functions described below are ignored.
- **EnableRecording = TRUE**
Make this true to have MediCollector record signals to a datafile.
- **EnableStreaming = TRUE**
Make this TRUE to have MediCollector stream data out via TCP/IP. NOTE: You must enable either RECORDING or STREAMING.
- **StreamingPort = 2057**
If you have enabled streaming, use this key to specify the TCP port.

- StreamingIP = "127.0.0.1"**
 If you have enabled streaming, use this key to specify the target IP address where data will be streamed. This key is only used if the streaming protocol = HL7, otherwise, the IP is always the localhost.
- StreamingProtocol = "HL7"**
 If you have enabled streaming, use this key to specify the protocol. It must be either "HL7" or "MediCollector Format"
- DisableLiveGraphs = TRUE**
 If you make this key TRUE, MediCollector will not show the graphs that show live data flowing across the screen. Instead, it will show a blank screen. If you do this, you can use other existing functions within MediCollector (e.g. the Signal Info window) to monitor the recording. Making this TRUE will reduce the load on your CPU. If this key is TRUE, then the key `signalsToDisplay` (see below) is ignored.
- DeviceName = "Philips Intellivue Series"**
 Use this key to specify which device you want to use within MediCollector. If the specified device name is not found, MediCollector will fire error code 7014 and exit.
- SignalNames = "HR,Pulse,Resp"**
 This is a comma-separated list of signals you want to record. If you specify `SignalNames = "-1"` then MediCollector will record ALL available signals. MediCollector will search the device for the specified signals and select all the specified signals that are found. This allows you to account for the varying names you might find on different devices. For example, you could specify "P,Press,Pressure,BP,NIBP" to account for the varying labels for a blood pressure signal. Recording will only begin if *at least one* of the signal names are found. If *none* of the specified signals names are found, MediCollector will throw error 7014 and exit. Upon successful connection, a status message will be written to the "automationlog" (code 8010) and also the name(s) of the signal(s) that are being recorded (code 8015).
- SignalsToDisplay = "HR,Pulse"**
 This is a comma-separated list of signals you want to display in the window after recording has launched. If none of your specified signals are available for display (or if this key is empty or doesn't exist), then a randomly selected signal will be shown on-screen. If `DisableLiveGraphs = TRUE` then this key is ignored and no signals are displayed.
- COMPort = "COM5"**
 This is the COM port you want MediCollector to use. If the specified COM port is not found, an error (code 7011) will fire and MediCollector will exit.
- SessionParameter0 = "MySessionID"**
SessionParameter1 = "MySubjectID"
 Here you can specify your session parameters. If you have configured MediCollector to have a different collection of session parameters (by using EDIT>PREFERENCES), then you can add/remove more lines to your INI file as needed. For example, you could add `SessionParameter2 = "MyBedNumber"` to the INI file. Just make sure the session parameters you specify in your INI match up with the session parameters in EDIT>PREFERENCES. If you are missing a required session parameter, MediCollector will fire error code 7015 and exit.
- PathToDataFile = "C:\Program Files\MediCollector BEDSIDE\data\aaa\bbb\test15.medi"**
 Here you can specify the path to the ".medi" file where you want MediCollector to store data. If this field is

empty or nonexistent, MediCollector will auto-generate a target filename. For example, if this field is blank and you enter "Subject01" and "Session01" as the values of your first two session parameters, it will automatically store your file in this location:

C:\Path_to_your_data_directory\Subject01\Session01\Timestamp_Subject01_Session01.medi

An error will be fired if the specified path can not be written (code 7001), or if a file already exists at the path (code 7002).

- **SplitFiles = 0**

This setting controls how new files are created. While recording, MediCollector can split the datafile to prevent files from getting excessively large.

- **SplitFiles = 0** ... This means that the datafile will never split.
- **SplitFiles = 1** ... This means that a new file will be created at midnight every day.
- **SplitFiles = 2** ... Not supported yet.

- **PathToAutomationLog = "C:\Program Files\MediCollector BEDSIDE\automationlog.log"**

Here you can specify the path to a special log file used for this automation interface. MediCollector will pass error and status messages related to the automation process back to your application via this log file, which we'll call the "*automationlog*" in order to differentiate it from the regular MediCollector log file. Error codes and status messages to be recorded into this log are specified on following page. If this field is blank or nonexistent, no automationlog will be created.

- **PathToSignalListFile = "C:\Program Files\MediCollector BEDSIDE\signlistfile.txt"**

Upon successful connection to your device, MediCollector will write the list of available signal names to file. Use this key to specify the path where you want to save this file. Your remote application could then use this list of signal names to prompt the user, etc. If this field is blank or nonexistent, the file will not be created.

- **SecondsBetweenRecordActiveMsgs = 15**

MediCollector will write a "heartbeat" status message to the automationlog. This will allow your third party application to confirm that MediCollector is still alive and connected and recording data. MediCollector will write the status message "Recording active" (code 8025) to the automationlog every x seconds. Using this key, you can specify how many seconds elapse between every "Recording active" message. Your remote application can then monitor the automationlog to make sure these message continually appear. If they stop appearing, you know something bad has happened. This will allow you to detect if MediCollector crashes or something.

AutomationLog:

MediCollector will write errors and status messages related to the automation process to this log file. Each row is delimited using the “|” character as described below:

< Code>|<Error or Status>|<Timestamp>|<Index>|<Error Text>|

Errors will begin numbering from 7000 and status messages will begin number from 8000. The <Index> field will increment for every row appended to the automationlog.

Here is an example of an error message:

7011|Error|2015.06.11 14:17:06.5281|1|The specified COM port could not be found|

Here is an example of a status message:

8015|Status|2015.06.11 14:17:07.0000|2|Signals selected: ICP,HR,RR|

Here are the entire list of error and status messages:

Error Code	Message
7000	Fatal Internal MediCollector Error: <i><details inserted here></i>
7001	Can not write to the path (PathToDataFile) specified in settings.ini.
7002	A file already exists at the path (PathToDataFile) specified in settings.ini.
7010	Unable to auto-start MediCollector.
7011	The specified COM port could not be found.
7012	Unable to find patient monitor on specified COM port.
7013	None of the specified signal names could be found on the specified device
7014	The device name specified in settings.ini could not be found or does not have a valid license
7015	The settings.ini file does not contain all of the required session parameter values.
7016	Unable to write signal list to the specified path.
7017	The PathToAutomationLog is invalid. Please check settings.ini file.
7018	You have specified that both EnableStreaming and EnableRecording are FALSE.

Status Code	Message
8000	MediCollector warning: <i><details inserted here></i>
8005	MediCollector launched
8010	Connection established
8015	Signals selected: abc, xyz, lmn, opq
8020	Session begun
8025	Session active
8030	Session ended
8031	Signal <xyz> has gone stale
8032	Signal <xyz> is active again (no longer stale)
8035	MediCollector exited
8037	Signals currently active: abc, xyz, lmn, opq