

BTL HL7 ECG Export Rev-0.1

Usage Notes

Date: 07 July 2025

Author: Densray Designs Private Limited

1. Introduction

This document describes how to use the **BTL HL7 ECG Export** library for the ECG2 device or any desktop application (Windows/Linux). Written in C, the library is designed for efficient memory usage, with no dynamic memory allocation, enabling use on microcontroller platforms (except for the libxml2 opensource library for xml parsing). The primary purpose of this library is to export diagnostic information in HL7 format, including a PDF report and diagnostics meta data.

Upon initialization, the library runs a background thread to handle the export process. This thread handles the export of PDF and diagnostics data, constructs the HL7 message, and communicates with the server.

2. Files and Compilation

HL7 Export Library Files

- **btIHI7EcgExport.c**: The main implementation file for parsing and managing export operations.
- **btIHI7EcgExport.h**: Header file to be included by the user application for interaction with the export functionality.
- **btIHI7XmlNg.c**: For parsing diagnostic/ examination meta data XML (BTL XML-NG) and utility functions..
- **btIHI7XmlNg.h**: Header file for above.
- **btIHI7XmlExtraData.c and btIHI7XmlExtraData.h**: For parsing meta data such as Order GUID etc from received HL7 work order saved in the system as XML to use while exporting examination results.

- **btlHI7ExpParseHI7.c and btlHI7ExpParseHI7.h** : For parsing the ACK message expected from the server on completion of export.
- **btlHI7ExpDebug.h**: For controlling debug logging.

Test Program

- **btlHI7EcgExportTestmain.c**: A test application to demonstrate how to use the export library.

Compilation

To compile, include all the `.c` and `.h` files in your project. On Windows, ensure that the `BTL_HL7_FOR_WINDOWS` macro is defined in the `btlHI7EcgExport.h`. (if it does not get automatically defined based on MACROs such as `_WIN32`. See `README_BTLHL7EXPORT.txt` for more detailed instructions. This pdf HL7 export library is fully independent of the HL7 work list server library.

3. Usage

The basic usage involves the following steps: (See `btlHI7EcgExportTestmain.c` and the `README_BTLHL7EXPORT.txt`)

1. **Allocate memory for the export structure** of type `BtlHI7Export_t`. All API calls are passed a pointer to this as the first argument.
2. **Initialize the export instance** using `btlHI7ExportInit()`, which also starts the background export thread . You can also pass IP address and port of the HL7 Export server while calling this function or you may set it later at any time .
3. **Set the server IP address and port** using `btlHI7ExpSetSrvIp()`, if not already set in step above. You may change the IP address and port any time using this function.
4. **Initiate the export by providing the paths** for the PDF report, diagnostics XML, and additional protocol data using `btlHI7ExportPdfReport()`. This function will return immediately and export will be taken up in this own thread. You must monitor the status and start a new export only after it completes successfully or aborts due to error.
5. **Monitor (poll) export status** using the `btlHI7GetExportStatus()` function.

6. Thread can be shutdown by calling `bt1HI7ExpShutdown()` and polling the status of the shutdown operation using `bt1HI7ExpIsThreadRunning()`.

3.1 Example Program

Check `bt1HI7EcgExportTestmain.c` for the latest version of the code.

```
#include "bt1HI7EcgExport.h"
```

```
// Configuration for the export
```

```
Bt1HI7Export_t gBt1HI7Export;
```

```
char gHI7ExpSrvIpAddrStr[16] = "127.0.0.1";
```

```
uint16_t gHI7ExpSrvPort = 23727;
```

```
// File paths for test
```

```
char gPdfFileName[256] = "hl7Test_1.pdf";
```

```
char gBt1XmlNgFileName[256] = "d60b513e-d780-4ba0-8e57-a66d2ff8d42b.diagnostics.xml";
```

```
// Protocol extra data (XML string)
```

```
char gExtraData[] = "<ProtocolExtraData Name=\"HL7\">\n<Segment  
id=\"MSH\">\n<HL7Version>2.3</HL7Version>\n</Segment>\n...</ProtocolExtraData>";
```

```
int main() {
```

```
    int expStatus;
```

```
    Bt1HI7Export_t* pHI7Exp = &gBt1HI7Export;
```

```
    // Initialize export instance
```

```
    bt1HI7ExportInit(pHI7Exp, gHI7ExpSrvIpAddrStr, gHI7ExpSrvPort);
```

```
    // Start the export with PDF, diagnostics XML, and protocol data
```

```
    int extraDataSize = strlen(gExtraData);
```

```
    expStatus = bt1HI7ExportPdfReport(pHI7Exp, gPdfFileName, gBt1XmlNgFileName, gExtraData,  
extraDataSize);
```

```
    if (expStatus != 0) {
```

```

    printf("Error: Export failed: %d\n", expStatus);

    return 1;
}

// Monitor the export status
while (1) {

    Sleep(2000); // Sleep for 2 seconds

    expStatus = btlHI7GetExportStatus(pHI7Exp);

    if (expStatus == BTLHL7EXP_STATUS_INPROGRESS) {

        printf("Export in progress...\n");

    } else if (expStatus == BTLHL7EXP_STATUS_COMPLETE) {

        printf("Export completed successfully.\n");

        break;

    } else {

        printf("Error occurred: %d\n", expStatus);

        break;

    }

}

return 0;
}

```

4. Key Functions

- *btlHI7ExportInit(BtlHI7Export_t pExport, char ipAddrStr, uint16_t port)*: Initializes the export module and starts the export thread.
- *btlHI7ExpSetSrvIp(BtlHI7Export_t pHI7Exp, char ipAddrStr, uint16_t port)*: Configures the server's IP address and port for communication.

- *btlHI7ExportPdfReport(BtlHI7Export_t pExport, char pdfPath, char* diagnosticsXmlPath, char* protocolExtraData, int protocolExtraDataLen)*: Starts the export of the PDF and diagnostics data.
- *btlHI7GetExportStatus(BtlHI7Export_t pHI7Exp)*: Retrieves the current status of the export operation.
- *btlHI7ExpSetServerTimeout(BtlHI7Export_t* pHI7Exp, int timeoutSeconds)*: Sets the server time out in seconds. The default is 30 seconds if this call is not invoked.
- *btlHI7ExpAbortExport(BtlHI7Export_t* pHI7Exp)*: Aborts the current export operation.
- *btlHI7ExpShutdown(BtlHI7Export_t* pHI7Exp)*: Shuts down the export thread. After this export operations if triggered will be ignored.
- *btlHI7StartExportThread(BtlHI7Export_t* pHI7Exp)*: Starts the export thread.
- *btlHI7ExpIsThreadRunning(BtlHI7Export_t* pHI7Exp)*: Returns 0 if thread is not running, else non-zero.

5. Error Handling

The export process may return various error codes (-ve numbers) to indicate issues. Some common errors are given below. See btlHI7EcgExport.h for a complete list of MACROs.

- **BTLHL7EXP_STATUS_ERR_SOCKET**: Socket creation failed.
- **BTLHL7EXP_STATUS_ERR_CONNECT**: Connection to the server failed.
- **BTLHL7EXP_STATUS_ERR_NETWORK**: Network error encountered.
- **BTLHL7EXP_STATUS_ERR_ACK_TIMEOUT**: Acknowledgement timeout.
- **BTLHL7EXP_STATUS_ERR_ACK**: Acknowledgement failure.
- **BTLHL7EXP_STATUS_INPUT_ERR**: Invalid input provided to the export function.
- **BTLHL7EXP_STATUS_EXDATA_XML_ERR**: Error parsing the protocol extra data XML.
- **BTLHL7EXP_STATUS_PROCESSING_ERR**: Error during export processing.

[End of document]

