# GDT Integration – Operation

## Table of Contents


CLI Reference . . . 1
- Usage . . . 1
- Parameters . . . 1
- Exit Codes (CLI Mode) . . . 2
- Examples . . . 2

Directory Structure During Operation . . . 3
Processing Workflow . . . 3
Concurrent Processing . . . 4
Error Handling . . . 4
HL7 Message Queue . . . 4
Logging . . . 5
Typical Console Output . . . 5


# CLI Reference

## Usage

```
gdt-hl7-bridge.exe [--daemon] [--studyuid <uid>] [--accnbr <nr>] [--patid <id>]
                   [--nachname <name>] [--vorname <name>]
                   [--geburtsdatum <DDMMYYYY>] [--fallnummer <nr>]
                   [--help]
```

## Parameters

| Parameter | Mode | Description |
|---|---|---|
| `--daemon` | Daemon | Starts the GDT polling service. Runs continuously in the background. |
| `--studyuid <uid>` | CLI | DICOM Study UID as the search parameter for PDF retrieval. |
| `--accnbr <nr>` | CLI | Order number / accession number as the search parameter. |
| `--patid <id>` | CLI | Patient number as the search parameter. |
| `--scope <level>` | CLI | Share level of the access code: `patient`, `accession`, `study` or `auto`. Defaults to `gdtImport.scope`, otherwise the portal setting applies. See Share Level. |

| Parameter | Mode | Description |
|---|---|---|
| `--profile <name>` | CLI | Profile whose settings are used. Required if several profiles are configured. See Profiles. |
| `--print-profiles` | Check | Prints all profiles as the bridge uses them (template and profile merged): input directory, GDT fields, search order, share level and all outputs. Values set in the profile are marked with *, the integration token is masked. With `--profile <name>`, only that profile. Does not call the portal and does not create directories. |
| `--nachname <name>` | CLI | Last name – optional, for filename templates and HL7 context. |
| `--vorname <name>` | CLI | First name – optional, for filename templates and HL7 context. |
| `--geburtsdatum <DDMMYYYY>` | CLI | Date of birth in GDT format – optional, automatically converted to YYYYMMDD for HL7. |
| `--fallnummer <nr>` | CLI | Case number – optional, for filename templates and HL7 (PV1 segment). |
| `--help` / `-h` | both | Show help and exit. Also shown when called without parameters. |

When multiple search parameters are provided, `gdtImport.priority` (default: `studyuid → accnbr → patid`) determines which one is used for the API call.

# Exit Codes (CLI Mode)

| Code | Meaning |
|---|---|
| `0` | Pipeline completed successfully |
| `1` | Error (no search parameter, HTTP error, MLLP timeout, etc.) |

# Examples

```
# Retrieve PDF by Study UID, send HL7 MDM
gdt-hl7-bridge.exe --studyuid 2.25.186760260011452398476108131496539831798

# Retrieve PDF by accession number, with patient data for HL7 context
gdt-hl7-bridge.exe --accnbr 186B7928 --patid 3209648408 --nachname Mueller --vorname
Max

# Call from a batch script with error check
gdt-hl7-bridge.exe --studyuid %STUDYUID%
if errorlevel 1 echo ERROR retrieving PDF
```

# Directory Structure During Operation

`gdt-hl7-bridge` manages three subdirectories below `gdtImport.gdtInputDir`:

```
gdtInputDir/
 ├──── *.gdt                          ← New input files (placed here by the practice
software)
 ├──── done/
 │     └──── *.gdt                     ← Successfully processed files
 └──── error/
     ├──── *_YYYYMMDD-HHmmss.gdt      ← Failed files (timestamp appended to filename)
     └──── *_YYYYMMDD-HHmmss.gdt.err  ← Error message as a text file (same base name)
```

The `done/` and `error/` subdirectories are created automatically when first needed.

# Processing Workflow

For each `*.gdt` file found, the following steps are executed:

1. **Stability check** – Files modified within the last 500 ms are skipped (protection against ongoing writes by the practice software).
2. **GDT parsing** – The input file is read in GDT 2.1 format (ISO-8859-1 encoding). Both compact format (without spaces) and standard format are supported.
3. **Field resolution** – All fields defined in `gdtImport.mapping` are extracted from the file.
4. **Required field check** – Missing required fields from `gdtImport.required` result in an error.
5. **Priority selection** – The first key from `gdtImport.priority` for which a value is found determines the API parameter.
6. **Share level** – The share level is determined from the GDT field `mapping.scope` or the profile's `scope` (see Share Level).
7. **PDF retrieval** – The PDF is downloaded from the portal at `/api/reports/pdf?session=…&<param>=<value>&scope=<level>`. HTTP redirects (301, 302, 307) are followed automatically. *(if `pdfFile.enabled`)*
8. **PDF saved** – The PDF is stored at `pdfFile.pdfDir/<pdfFileName>`. *(if `pdfFile.enabled`)*
9. **GDT response written** – The response file is created at `gdtWriter.gdtOutputDir/<gdtFileName>`. *(if `gdtWriter.enabled`)*
10. **HL7 MDM^T02 sent/saved** – The HL7 message is sent via MLLP/TCP *(if `hl7Sender.enabled`)* and/or saved as a file *(if `hl7Writer.enabled`)*. If sending fails and `hl7Sender.queue.enabled` is set, the message is written to the queue instead of aborting the processing.
11. **Cleanup** – The input file is moved to `done/`.

# Concurrent Processing

Multiple GDT files in the input directory are processed concurrently (in parallel). Each file is tracked internally as a separate job; a file already being processed is skipped in the next poll cycle.

# Error Handling

If an error occurs during processing (e.g. HTTP error when retrieving the PDF, missing required field, MLLP timeout, invalid GDT file), the following happens:

1. The error message is printed to the console.
2. A text file `<filename>.gdt.err` containing the error message is created in `error/`.
3. The input file is moved to `error/<filename>.gdt`.

*Example after an HTTP 400 error*

```
error/
 ├──── input-accnbr_20260701-141203.gdt
 └──── input-accnbr_20260701-141203.gdt.err   ← Content: "HTTP 400 von
https://portal.../api/reports/pdf?session=*** [STUDY_NOT_FOUND]: The requested study
was not found."
```

*Example after an MLLP timeout*

```
error/
 ├──── input_20260701-141203.gdt
 └──── input_20260701-141203.gdt.err   ← Content: "MLLP timeout after 10000ms
connecting to ris-server:2333"
```

| NOTE | The session token in the portal URL is automatically masked as * in all log output and error files. |
|---|---|

# HL7 Message Queue

When `hl7Sender.queue.enabled: true` is configured, HL7 messages are not lost on connection failure.

| Situation | Behavior |
|---|---|
| Receiver unreachable (ECONNREFUSED, timeout) | Message is saved as a `.hl7` file in `queue.dir`. Processing is considered successful — the GDT file is moved to `done/`. |
| Daemon mode | Every `queue.retryIntervalMs` milliseconds (default: 60 s), the daemon attempts to drain the queue. On the first failure, it stops and retries at the next interval. |

| Situation | Behavior |
|---|---|
| CLI mode | After processing, the queue is drained once immediately. The next invocation repeats this. |
| Receiver becomes reachable again | Queued messages are resent in chronological order and removed from the queue after a successful ACK. |

```
queue/hl7/
 ├──── 1751111295123-a4f2b.hl7   ← pending message
 └──── 1751111301456-c9d1e.hl7   ← pending message
```

# Logging

The application writes all messages to standard output (`stdout`). Additionally, the `log` module in `config.json` can enable a daily rotating log file — see Module: log.

The log file rotates daily and contains timestamps:

```
2026-06-28 13:48:15 [INFO] gdt-hl7-bridge started [--daemon]
2026-06-28 13:48:15 [INFO]   Input:    C:\RadioWeb\GDT\Input
2026-06-28 14:12:03 [INFO] Processing: input.gdt
2026-06-28 14:12:03 [INFO]   HL7:    send failed (connect ECONNREFUSED ...) — queued
as 1751111295123-a4f2b.hl7
2026-06-28 14:13:03 [INFO] Queue: 1 message(s) pending — retrying ris-server:2333
2026-06-28 14:13:03 [INFO] Queue: sent and removed 1751111295123-a4f2b.hl7
```

**TIP** The `log` module handles rotation and retention automatically. Manual redirection of console output is no longer necessary.

# Typical Console Output

*Daemon mode – normal operation*

```
gdt-hl7-bridge started [--daemon]
  Input:     C:\RadioWeb\GDT\Input
  GDT Out:   C:\RadioWeb\GDT\Output
  PDFs:      C:\RadioWeb\PDFs
  HL7 TCP:   ris-server:2333
  HL7 Queue: queue/hl7 (retry 60000ms)
  HL7 Out:   C:\RadioWeb\HL7\Output
  Interval:  2000 ms
Processing: input.gdt
  Lookup: patid=3209648408
  PDF:    C:\RadioWeb\PDFs\3209648408__1719475200000.pdf
  GDT:    C:\RadioWeb\GDT\Output\3209648408__1719475200000.gdt
```

```
  HL7:    sending to ris-server:2333
  HL7:    sent
  HL7:    C:\RadioWeb\HL7\Output\3209648408__1719475200000.hl7
  Done
```

*Daemon mode – receiver unreachable, queue active*

```
Processing: input.gdt
  Lookup: patid=3209648408
  HL7:    sending to ris-server:2333
  HL7:    send failed (connect ECONNREFUSED 10.0.0.1:2333) — queued as 1751111295123-
a4f2b.hl7
  Done
Queue: 1 message(s) pending — retrying ris-server:2333
Queue: sent and removed 1751111295123-a4f2b.hl7
```

*CLI mode*

```
gdt-hl7-bridge [CLI]
  Lookup: studyuid=2.25.186760260011452398476108131496539831798
  PDF:    C:\RadioWeb\PDFs\3209648408__1719475200000.pdf
  HL7:    sending to ris-server:2333
  HL7:    sent
Done.
```