HL7 MDM^T02

Table of Contents

Transmission Modes . . . 1
Message Format . . . 1
MLLP Protocol . . . 2
Configuration . . . 2
  Example . . . 2
  Fields . . . 3
Template (`mdm-output.template`) . . . 3
  Default Template . . . 3
  Template Variables . . . 4
Testing the Connection . . . 4

After each processed GDT request, the `hl7Sender` module sends an HL7 message of type **MDM^T02** (*Medical Document Management – Original Document Notification*). The message contains the PDF report as a Base64-encoded attachment as well as DICOM metadata from the incoming GDT request.

This allows downstream systems (HIS, RIS) to receive the report automatically without relying on the GDT response protocol.

Transmission Modes

Both output channels can be enabled independently and can be active at the same time.

Mode	Activation	Description
MLLP/TCP	`hl7Sender.enabled: true`	The message is sent to an HL7 listener via the MLLP protocol. Waits for an ACK (MSA segment), timeout 10 seconds.
File output	`hl7Writer.enabled: true`	The message is stored as a UTF-8 text file in `hl7Writer.hl7OutputDir`. Useful for testing or downstream file-based processing.

Message Format

The message conforms to HL7 v2.3.1, message type MDM^T02:

Segment	Content
`MSH`	Message header – sender `RADIOWEB|PORTAL`, timestamp, message ID

Segment	Content
`EVN`	Event timestamp
`PID`	Patient data (ID, name, date of birth)
`PV1`	Visit information (case number)
`TXA`	Document announcement (type `RAD`, status `AU|AV`)
`OBX[1]`	PDF report as Base64 attachment (`ED` type, MIME `AP/PDF`)
`OBX[2]`	Document description (modality, study description from `vars`)

MLLP Protocol

MLLP (*Minimum Lower Layer Protocol*) is the standard transport protocol for HL7 messages over TCP. `gdt-hl7-bridge` frames every message as follows:

```
0x0B  ← Start Block (Vertical Tab)
<HL7 message as UTF-8 text>
0x1C  ← End Data (File Separator)
0x0D  ← Carriage Return
```

The receiver's MLLP listener must understand this framing (standard in all common HL7 engines).

Configuration

Example

```
"hl7Sender": {
  "enabled":      true,
  "templateFile": "templates/mdm-output.template",
  "host":         "ris-server",
  "port":         2333,
  "queue": {
    "enabled":         true,
    "dir":             "queue/hl7",
    "retryIntervalMs": 60000
  }
},

"hl7Writer": {
  "enabled":      false,
  "templateFile": "templates/mdm-output.template",
  "hl7OutputDir": "C:\\RadioWeb\\HL7\\Output",
  "hl7FileName":  "{{patid}}_{{accnbr}}_{{timestamp}}.hl7"
}
```

Fields

`hl7Sender` – **MLLP/TCP transmission**

Field	Required	Default	Description
`enabled`	no	`false`	Enable MLLP/TCP transmission.
`templateFile`	yes (if enabled)	–	Path to the Handlebars template for the MDM^T02 message (relative to the application directory or absolute).
`host`	yes (if enabled)	–	Host name or IP address of the HL7 receiver (MLLP listener).
`port`	yes (if enabled)	–	TCP port of the MLLP listener.
queue – message queue on transmission errors			
`queue.enabled`	no	`false`	Enable the queue. If transmission fails (connection refused, timeout), the message is stored in `queue.dir` instead of being lost.
`queue.dir`	yes (if queue.enabled)	–	Directory for buffered HL7 messages. Created automatically. In daemon mode the queue is flushed periodically as soon as the receiver is reachable again.
`queue.retryIntervalMs`	no	`60000`	Interval in milliseconds at which the daemon tries to flush the queue. Only active in daemon mode.

`hl7Writer` – **file output**

Field	Required	Default	Description
`enabled`	no	`false`	Enable file output. Independent of `hl7Sender`.
`templateFile`	yes (if enabled)	–	Path to the Handlebars template (can be identical to `hl7Sender.templateFile`).
`hl7OutputDir`	yes (if enabled)	–	Output directory for HL7 files. Created automatically if needed.
`hl7FileName`	yes (if enabled)	–	File name of the HL7 output file. Handlebars template, see Template Variables.

Template (`mdm-output.template`)

The template controls the exact structure of the MDM^T02 message and uses Handlebars syntax.

Default Template

```
MSH|^~\&|RADIOWEB|PORTAL|SCN|RIS|{{now}}||MDM^T02|{{msgId}}|P|2.3.1|
EVN||{{now}}
PID|||{{patientId}}||||{{patientName}}||{{birthDate}}
PV1||O|||||||||||||||||{{fallnr}}
TXA|1|RAD||{{now}}|||||||||{{docId}}||||{{filename}}|AU||AV
OBX|1|ED|18748-4^Diagnostic Imaging
Report^LN||qr2mdm^AP^PDF^Base64^{{pdfBase64}}||||||F|||||||{{studyUid}}
OBX|2|ST|^Document Description||VEPRO QR Code {{modality}} {{studyDescription}}
```

Template Variables

Variable	Source	Content
`{{now}}`	automatic	HL7 timestamp in the format `YYYYMMDDHHmmss` (e.g. `20260628143022`)
`{{msgId}}`	automatic	Unique message ID (Base-36-encoded Unix timestamp)
`{{patientId}}`	GDT field 3000	Patient number
`{{patientName}}`	GDT fields 3101, 3102	Name in HL7 format `LastName^FirstName`
`{{birthDate}}`	GDT field 3103	Date of birth in HL7 format `YYYYMMDD` (converted automatically from GDT format DDMMYYYY)
`{{fallnr}}`	GDT field 8402	Case number
`{{studyUid}}`	GDT field 8410	DICOM Study UID
`{{docId}}`	GDT field 8402 or 3000	Document ID (order number, fallback: patient number)
`{{filename}}`	automatic	File name of the locally stored PDF (without path)
`{{pdfBase64}}`	automatic	PDF file, Base64-encoded. The PDF is always downloaded when `hl7Sender` or `hl7Writer` is active – regardless of `pdfFile.enabled`.
`{{<key>}}`	`vars`	All variables from `config.json` → `vars`, e.g. `{{modality}}`, `{{studyDescription}}`

NOTE `{{modality}}` and `{{studyDescription}}` cannot be derived from the GDT request and must be set via `vars` in the configuration if needed.

NOTE GDT encodes dates of birth in the format `DDMMYYYY`; HL7 expects `YYYYMMDD`. The conversion for `{{birthDate}}` happens automatically.

Testing the Connection

A simple TCP listener can be used to test the MLLP receiver:

```
# Linux/macOS
nc -l 2333

# Windows (PowerShell, requires ncat or similar)
ncat -l 2333
```

After processing a GDT test file, the MLLP-framed HL7 message appears in the terminal. The listener does not need to send an ACK – `gdt-hl7-bridge` closes the connection after receiving `FS CR` or when the connection is closed.