GDT Integration – Configuration

Table of Contents

Full Example . . . 1
Connection . . . 3
Module: `pdfFile` . . . 3
Module: `gdtImport` . . . 3
Module: `gdtWriter` . . . 4
Module: `hl7Sender` . . . 4
Module: `hl7Writer` . . . 5
Module: `log` . . . 5
`vars` – Global Template Variables . . . 6
Field Mapping (`gdtImport.mapping`) . . . 6
Required Fields (`gdtImport.required`) . . . 7
Priority (`gdtImport.priority`) . . . 7
Share Level (`gdtImport.scope`) . . . 7
Profiles (`profiles`) . . . 8
Template Variables . . . 10
GDT Response Template . . . 11
- Default Template . . . 11
- Template Context . . . 11
- Generated GDT Fields . . . 12

All configuration for `gdt-hl7-bridge` is done via the file `config\config.json`. A template with all available fields is provided as `config\config.json.example`.

The configuration is modular: each functional block is controlled by its own object with an `enabled` flag. Unused modules can be disabled without removing their configuration.

Full Example

```
{
  "portalUrl": "https://portal.your-practice.com",
  "apiToken":  "XXXX-XXXX-XXXX",

  "pdfFile": {
    "enabled":     true,
    "pdfDir":      "C:\\RadioWeb\\PDFs",
    "pdfFileName": "{{patid}}_{{nachname}}_{{vorname}}_{{accnbr}}_{{timestamp}}.pdf"
  },
```

```
  "gdtImport": {
    "enabled":       true,
    "gdtInputDir":   "C:\\RadioWeb\\GDT\\Input",
    "pollIntervalMs": 2000,
    "mapping": {
      "patid":        "3000",
      "nachname":     "3101",
      "vorname":      "3102",
      "geburtsdatum": "3103",
      "accnbr":       "8402",
      "fallnummer":   "8402",
      "studyuid":     "8410"
    },
    "required": ["patid"],
    "priority":  ["studyuid", "accnbr", "patid"]
  },

  "gdtWriter": {
    "enabled":      true,
    "templateFile": "templates/gdt-output.template",
    "gdtOutputDir": "C:\\RadioWeb\\GDT\\Output",
    "gdtFileName":  "{{patid}}_{{accnbr}}_{{timestamp}}.gdt",
    "senderId":     "RadioWeb"
  },

  "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"
  },

  "log": {
    "enabled":  true,
    "level":    "info",
    "file":     "logs/gdt-hl7-bridge-%DATE%.log",
    "maxSize":  "10m",
    "maxFiles": "30d"
  },
```

```
  "vars": {
    "institute": "Dres. Mueller Schmidt",

    "doc6302": "",
    "doc6304": "Radiological Report",

    "SF2":  "Dres. Mueller Schmidt",
    "SF3":  "1 Example Street",
    "SF6":  "00000 Anytown",
    "SF7":  "Germany",
    "SF16": "+49 0815 4711"
  }
}
```

Connection

Field	Required	Description
`portalUrl`	yes	Base URL of the Patient Portal, e.g. `https://portal.your-practice.com`. No trailing slash.
`apiToken`	yes	Customer integration token. Generated in the portal admin under *Settings* → *Integration* and validated server-side.

Module: `pdfFile`

Controls the download and local storage of the PDF report.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module. If `false`, no PDF is downloaded.
`pdfDir`	yes (if enabled)	–	Directory where downloaded PDF reports are stored. Created automatically when needed.
`pdfFileName`	yes (if enabled)	–	Filename for the saved PDF. Handlebars template, see Template Variables.

Module: `gdtImport`

Controls monitoring of the GDT input directory and reading of GDT fields.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module. Without this module, the application does not monitor any directory.

Field	Required	Default	Description
`gdtInputDir`	yes (if enabled)	–	Directory monitored for incoming `*.gdt` files. Created automatically on startup.
`pollIntervalMs`	no	`2000`	Polling interval in milliseconds.
`mapping`	yes (if enabled)	–	Object: mapping key → GDT field ID. See Field Mapping (`gdtImport.mapping`).
`required`	yes (if enabled)	–	Array of mapping keys that must be present in the input file.
`priority`	yes (if enabled)	–	Array of mapping keys in descending priority for the portal API call.
`scope`	no	portal setting	Share level of the created access code: `patient`, `accession`, `study` or `auto`. See Share Level (`gdtImport.scope`).
`scopeValues`	no	–	Translates values of the GDT field `mapping.scope` into share levels, e.g. `{"P": "patient", "A": "accession"}`.

Module: `gdtWriter`

Generates a GDT response file (record type 6310) with the local PDF path.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module.
`templateFile`	no	`templates/gdt-output.template`	Path to the Handlebars template (relative to the application directory or absolute).
`gdtOutputDir`	yes (if enabled)	–	Directory for GDT response files. Created automatically when needed.
`gdtFileName`	yes (if enabled)	–	Filename for the GDT response file. Handlebars template, see Template Variables.
`senderId`	no	`RadioWeb`	Identifier of the sending application in the GDT response file (field 8316).

Module: `hl7Sender`

Sends an HL7 MDM^T02 message via MLLP/TCP to an HL7 receiver.

For a complete description of the HL7 message format, see HL7 MDM^T02.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module.

Field	Required	Default	Description
`templateFile`	yes (if enabled)	–	Path to the Handlebars template for the HL7 message.
`host`	yes (if enabled)	–	Hostname or IP address of the MLLP listener.
`port`	yes (if enabled)	–	TCP port of the MLLP listener.
queue – Message queue on send failure			
`queue.enabled`	no	`false`	Enable the queue. On connection failure, the message is stored in `queue.dir` instead of being lost. Full description: HL7 Message Queue.
`queue.dir`	yes (if queue.enabled)	–	Directory for queued HL7 messages. Created automatically.
`queue.retryIntervalMs`	no	`60000`	Interval in milliseconds at which the daemon drains the queue. In CLI mode, the queue is drained immediately on each invocation.

Module: `hl7Writer`

Writes the HL7 MDM^T02 message as a text file to a local directory. Useful for testing or downstream file-based processing. Can be enabled independently of `hl7Sender`.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module.
`templateFile`	yes (if enabled)	–	Path to the Handlebars template for the HL7 message.
`hl7OutputDir`	yes (if enabled)	–	Output directory for HL7 files. Created automatically when needed.
`hl7FileName`	yes (if enabled)	–	Filename for the HL7 output file. Handlebars template, see Template Variables.

Module: `log`

Writes all runtime output to a daily rotating log file. Console output remains active — this module supplements it with a persistent log file.

Field	Required	Default	Description
`enabled`	no	`false`	Enable file logging.
`level`	no	`info`	Minimum log level: `error`, `warn`, `info`, `verbose`, `debug`.

Field	Required	Default	Description
`file`	no	`logs/gdt-hl7-bridge-%DATE%.log`	Path to the log file (relative to the application directory or absolute). `%DATE%` is replaced with the current date (`YYYY-MM-DD`). The directory is created automatically.
`maxSize`	no	`10m`	Maximum file size per log file. Units: `k` (KB), `m` (MB), `g` (GB).
`maxFiles`	no	`30d`	Retention period. Suffix `d` = days (e.g. `30d`), without suffix = maximum number of files (e.g. `14`).

`vars` – Global Template Variables

The `vars` object defines freely configurable variables available in *all* templates (GDT and HL7). Typical use cases are practice or institution data that rarely changes.

```
"vars": {
  "institute": "Dres. Mueller Schmidt",
  "doc6302": "",
  "doc6304": "Radiological Report",
  "SF2":  "Dres. Mueller Schmidt",
  "SF3":  "1 Example Street",
  "SF6":  "00000 Anytown",
  "SF7":  "Germany",
  "SF16": "+49 0815 4711"
}
```

Keys can be freely chosen and are accessible in templates directly as `{{institute}}`, `{{SF2}}`, etc.

Two keys have a special role in the default GDT response template:

Key	Meaning
`doc6302`	GDT field 6302 (report request marker). If empty or not set, the field is omitted.
`doc6304`	GDT field 6304 (document title). If empty or not set, the field is omitted.

Field Mapping (`gdtImport.mapping`)

The `mapping` object maps logical identifiers (mapping keys) to the corresponding GDT field IDs. All mapped fields are read from the input file and are available as template variables.

Key	GDT Field ID	Meaning
`patid`	`3000`	Patient number

Key	GDT Field ID	Meaning
`nachname`	`3101`	Patient last name
`vorname`	`3102`	Patient first name
`geburtsdatum`	`3103`	Date of birth (GDT format DDMMYYYY)
`accnbr`	`8402`	Order number / accession number
`fallnummer`	`8402`	Case number (alias for `accnbr`, same field ID)
`studyuid`	`8410`	DICOM Study UID (field assignment is vendor-dependent)
`scope`	any (optional)	Share level per request, see Share Level (`gdtImport.scope`). Only set it if the practice software can fill a free field.

NOTE GDT field IDs can be adjusted if the sending practice software uses different field IDs.

Required Fields (`gdtImport.required`)

The `required` array contains the mapping keys that must be present in the input file. If a required field is missing, the file is moved to `error/` with an error message.

```
"required": ["patid"]
```

Priority (`gdtImport.priority`)

The `priority` array determines which value is used for the API call when multiple mapped fields are present. Exactly the first key for which a value is found in the input file is used.

```
"priority": ["studyuid", "accnbr", "patid"]
```

Table 1. Examples

Fields present in GDT	Parameter used	Query
`studyuid` + `patid`	`studyuid`	`?studyuid=2.25.186…`
`accnbr` + `patid`	`accnbr`	`?accnbr=186B7928`
`patid` only	`patid`	`?patid=3209648408`

Share Level (`gdtImport.scope`)

When a PDF is requested, the portal creates an access code. The share level determines which studies this access code shares.

Value	Shared studies
`patient`	All studies of the patient (patient share).
`accession`	All studies of the accession number. When requested by `studyuid`, these are all studies with the same accession number as that study.
`study`	Only the requested studies (study share): exactly that study when requested by `studyuid`, all studies of the accession number when requested by `accnbr`.
`auto`	`accession` if the GDT file contains an accession number or Study UID, otherwise `patient`.

The level is determined in this order:

1. GDT field `mapping.scope`, if it is filled in the input file. Values are translated via `scopeValues`, otherwise used as they are.
2. `gdtImport.scope` (in the profile or the template, see Profiles (`profiles`)).
3. Without either, the portal setting **PDF Access Code Share Level** (`token.pdf.max_scope`, default `PATIENT`) applies.

The portal setting is also the widest level allowed: if it is set to `ACCESSION`, for example, the portal rejects requests with `patient`.

In the following cases, the input file is moved to `error/` together with an `.err` file, and no access code is created:

- invalid value in the GDT field,
- `accession` or `study` without accession number and Study UID in the file (`SHARE_SCOPE_REQUIRES_STUDY`),
- level wider than allowed by the portal (`SHARE_SCOPE_NOT_ALLOWED`).

In CLI mode, the level is passed with `--scope patient|accession|study|auto`.

Share level via a GDT field (e.g. 6200)

```
"mapping": {
  "patid":  "3000",
  "accnbr": "8402",
  "scope":  "6200"
},
"scopeValues": { "P": "patient", "A": "accession", "S": "study" },
"scope": "auto"
```

Profiles (`profiles`)

With profiles, one bridge installation runs several independent flows, e.g. one per button in the practice software. Each profile has its own input directory and can configure all modules

differently, e.g. the share level, a different HL7 port or an additional `hl7Writer`.

- **Without `profiles`**, the whole configuration is the only profile (`default`). Existing configurations keep working unchanged.
- **With `profiles`**, everything outside `profiles` is only a template and does not run itself. Only the profiles run.
- A profile contains only its differences from the template:
  - Objects are merged, i.e. `"hl7Sender": { "port": 2334 }` only changes the port.
  - Arrays and single values replace the template value (e.g. `priority`).
  - `null` removes a template value.
- `"enabled": false` directly in the profile disables the whole profile.
- `log` is global and must not appear in profiles. Log lines contain the profile name, e.g. `Processing [accession]: …`.
- `portalUrl` and `apiToken` can differ per profile, e.g. for a second practice with its own integration token.
- Keys starting with `_` are comments and are ignored.

```
{
  "portalUrl": "https://portal.your-practice.com",
  "apiToken":  "XXXX-XXXX-XXXX",
  "log":       { "enabled": true, "level": "info", "file": "logs/gdt-hl7-bridge-
%DATE%.log" },

  "pdfFile":   { "enabled": true, "pdfDir": "C:\\RadioWeb\\PDFs", "pdfFileName":
"{{patid}}_{{accnbr}}_{{timestamp}}.pdf" },
  "gdtImport": {
    "enabled":  true,
    "mapping":  { "patid": "3000", "accnbr": "8402", "studyuid": "8410" },
    "required": ["patid"],
    "priority": ["studyuid", "accnbr", "patid"]
  },
  "gdtWriter": { "enabled": true, "gdtOutputDir": "C:\\RadioWeb\\GDT\\Output",
"gdtFileName": "{{patid}}_{{timestamp}}.gdt" },
  "hl7Sender": {
    "enabled": false, "templateFile": "templates/mdm-output.template", "host": "ris-
server", "port": 2333,
    "queue": { "enabled": true, "dir": "queue/hl7" }
  },
  "hl7Writer": { "enabled": false, "templateFile": "templates/mdm-output.template",
"hl7OutputDir": "C:\\RadioWeb\\HL7\\Output", "hl7FileName":
"{{patid}}_{{timestamp}}.hl7" },

  "profiles": {
    "patient": {
      "gdtImport": { "gdtInputDir": "C:\\RadioWeb\\GDT\\Patient", "scope": "patient" }
    },
```

```
    "accession": {
      "gdtImport": { "gdtInputDir": "C:\\RadioWeb\\GDT\\Accession", "scope":
"accession" },
      "hl7Sender": { "enabled": true, "port": 2334, "queue": { "dir": "queue/hl7-
accession" } },
      "hl7Writer": { "enabled": true }
    }
  }
}
```

In the example, `patient` creates a patient share and a GDT response. `accession` only shares the accession number, additionally sends HL7 to port 2334 and also stores the HL7 message as a file.

Rules checked at startup

- All required fields are checked per profile (after merging). Error messages name the profile, e.g. `"profiles.accession.hl7Sender.host" missing`.
- Each profile needs its own `gdtImport.gdtInputDir`.
- Profiles whose HL7 messages go to different receivers (`host:port`) each need their own `hl7Sender.queue.dir`. The queue of a directory is always delivered to exactly one receiver.

In CLI mode, select the profile with `--profile <name>`. If there is only one profile, it is not required.

TIP After every change, run `gdt-hl7-bridge --print-profiles` to check what each profile actually does. The output shows the merged values and marks values coming from the profile with `*`.

Template Variables

The following variables are available in filename templates (`pdfFileName`, `gdtFileName`, `hl7FileName`). All keys defined in `mapping` are also available, provided the corresponding field is present in the input file.

Variable	Content
`{{patid}}`	Patient number (GDT field 3000)
`{{nachname}}`	Last name (GDT field 3101)
`{{vorname}}`	First name (GDT field 3102)
`{{geburtsdatum}}`	Date of birth in format DDMMYYYY (GDT field 3103)
`{{accnbr}}`	Order number / accession number (GDT field 8402)
`{{fallnummer}}`	Case number – identical to `accnbr` (GDT field 8402)
`{{studyuid}}`	DICOM Study UID (GDT field 8410, vendor-dependent)
`{{scope}}`	Determined share level (`patient`, `accession`, `study`; empty if the portal setting applies)
`{{timestamp}}`	Unix timestamp in milliseconds

Variable	Content
`{{date}}`	Current date in format `YYYY-MM-DD`
`{{time}}`	Current time in format `HH-MM-SS`
`{{inputFile}}`	Input filename without extension
`{{<key>}}`	Any variable from `vars` is available directly under its key name

GDT Response Template

The template `templates\gdt-output.template` controls the structure of the generated GDT response file in GDT 2.1 format. It uses Handlebars syntax.

The helper `{{gdtLine "FFFF" "content"}}` generates a correctly formatted GDT line including length prefix and CRLF. The length is calculated automatically: 3 (length field) + 4 (field ID) + content length + 2 (CRLF).

Default Template

```
{{~gdtLine "8000" "6310"~}}
{{~gdtLine "9206" "2.10"~}}
{{~gdtLine "8316" senderId~}}
{{~#if (lookup fields "8316")~}}{{gdtLine "8315" (lookup fields "8316")}}{{~/if~}}
{{~#if (lookup fields "8402")~}}{{gdtLine "8310" (lookup fields "8402")}}{{~/if~}}
{{~#if (lookup fields "3000")~}}{{gdtLine "3000" (lookup fields "3000")}}{{~/if~}}
{{~#if (lookup fields "3101")~}}{{gdtLine "3101" (lookup fields "3101")}}{{~/if~}}
{{~#if (lookup fields "3102")~}}{{gdtLine "3102" (lookup fields "3102")}}{{~/if~}}
{{~#if (lookup fields "3103")~}}{{gdtLine "3103" (lookup fields "3103")}}{{~/if~}}
{{~#if (lookup fields "3104")~}}{{gdtLine "3104" (lookup fields "3104")}}{{~/if~}}
{{~#if pdfPath~}}{{~#if doc6302~}}{{gdtLine "6302" doc6302}}{{~/if~}}{{gdtLine "6303"
"PDF"}}{{~#if doc6304~}}{{gdtLine "6304" doc6304}}{{~/if~}}{{gdtLine "6305"
pdfPath}}{{~/if~}}
```

Template Context

Variable	Content
`fields`	All GDT fields from the input file, accessible by field ID: `(lookup fields "3000")`
`pdfPath`	Absolute local path to the downloaded PDF file (empty if `pdfFile.enabled: false`)
`senderId`	Value from `gdtWriter.senderId`
`doc6302`	Value from `vars.doc6302` – optional report request marker (field 6302)
`doc6304`	Value from `vars.doc6304` – optional document title (field 6304)
`{{<key>}}`	All further variables from `vars`

Generated GDT Fields

Field ID	Meaning
8000	Record type: 6310 (examination data)
9206	GDT version: 2.10
8316	Sender ID (senderId)
8315	Receiver ID – taken from input field 8316 (practice software identifier), only if present
8310	Order number of the sender – taken from input field 8402 (accession number), only if present
3000 / 3101 / 3102 / 3103 / 3104	Patient data from the input file (only if present)
6302	Report request marker (from vars.doc6302, optional)
6303	File format: PDF (always set when pdfPath is present)
6304	Document title (from vars.doc6304, optional)
6305	File reference – local path to the PDF file