GDT-Integration – Konfiguration

Table of Contents

Vollständiges Beispiel . . . 1
Verbindung . . . 3
Modul: `pdfFile` . . . 3
Modul: `gdtImport` . . . 4
Modul: `gdtWriter` . . . 4
Modul: `hl7Sender` . . . 5
Modul: `hl7Writer` . . . 5
Modul: `log` . . . 6
`vars` – Globale Template-Variablen . . . 6
Feldmapping (`gdtImport.mapping`) . . . 7
Pflichtfelder (`gdtImport.required`) . . . 7
Priorität (`gdtImport.priority`) . . . 7
Freigabeebene (`gdtImport.scope`) . . . 8
Profile (`profiles`) . . . 9
Template-Variablen . . . 11
GDT-Antwort-Template . . . 11
- Standard-Template . . . 11
- Template-Kontext . . . 12
- Erzeugte GDT-Felder . . . 12

Die gesamte Konfiguration von `gdt-hl7-bridge` erfolgt über die Datei `config\config.json`. Eine Vorlage mit allen verfügbaren Feldern liegt als `config\config.json.example` bei.

Die Konfiguration ist modular aufgebaut: Jeder Funktionsblock wird durch ein eigenes Objekt mit einem `enabled`-Flag gesteuert. Nicht benötigte Module können deaktiviert werden, ohne die Konfiguration entfernen zu müssen.

Vollständiges Beispiel

```
{
  "portalUrl": "https://portal.ihre-praxis.de",
  "apiToken":  "Kann die Praxissoftware kein Feld für die Freigabeebene befüllenKann
die Praxissoftware kein Feld für die Freigabeebene befüllen-Kann die Praxissoftware
kein Feld für die Freigabeebene befüllenKann die Praxissoftware kein Feld für die
Freigabeebene befüllen-Kann die Praxissoftware kein Feld für die Freigabeebene
befüllenKann die Praxissoftware kein Feld für die Freigabeebene befüllen",
```

```
"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. Müller Schmidt",

    "doc6302": "",
    "doc6304": "Radiologischer Befund",

    "SF2":  "Dres. Müller Schmidt",
    "SF3":  "Albrechtstraße 1",
    "SF6":  "00000 Musterstadt",
    "SF7":  "Germany",
    "SF16": "+49 0815 4711"
  }
}
```

Verbindung

Feld	Pflicht	Beschreibung
`portalUrl`	ja	Basis-URL des Patientenportals, z.B. `https://portal.ihre-praxis.de`. Ohne abschließenden Schrägstrich.
`apiToken`	ja	Integrations-Token des Kunden. Wird im Portal-Admin unter *Einstellungen* → *Integration* erzeugt und serverseitig validiert.

Modul: `pdfFile`

Steuert den Download und die lokale Speicherung des PDF-Befunds.

Feld	Pflicht	Standard	Beschreibung
`enabled`	nein	`false`	Modul aktivieren. Wenn `false`, wird kein PDF heruntergeladen.
`pdfDir`	ja (wenn enabled)	–	Verzeichnis, in das heruntergeladene PDF-Befunde gespeichert werden. Wird bei Bedarf automatisch angelegt.
`pdfFileName`	ja (wenn enabled)	–	Dateiname der gespeicherten PDF. Handlebars-Template, siehe Template-Variablen.

Modul: `gdtImport`

Steuert die Überwachung des GDT-Eingangsverzeichnisses und das Einlesen der GDT-Felder.

Feld	Pflicht	Standard	Beschreibung
`enabled`	nein	`false`	Modul aktivieren. Ohne dieses Modul überwacht die Anwendung kein Verzeichnis.
`gdtInputDir`	ja (wenn enabled)	–	Verzeichnis, das auf eingehende `*.gdt`-Dateien überwacht wird. Wird beim Start automatisch angelegt.
`pollIntervalMs`	nein	`2000`	Prüfintervall in Millisekunden.
`mapping`	ja (wenn enabled)	–	Objekt: Mapping-Schlüssel → GDT-Feldkennung. Beschreibung: Feldmapping (`gdtImport.mapping`).
`required`	ja (wenn enabled)	–	Array der Mapping-Schlüssel, die in der Eingangsdatei zwingend vorhanden sein müssen.
`priority`	ja (wenn enabled)	–	Array der Mapping-Schlüssel in absteigender Priorität für den Portal-API-Aufruf.
`scope`	nein	Portal-Einstellung	Freigabeebene des erzeugten Zugangscodes: `patient`, `accession`, `study` oder `auto`. Siehe Freigabeebene (`gdtImport.scope`).
`scopeValues`	nein	–	Übersetzt Werte des GDT-Felds `mapping.scope` in Freigabeebenen, z.B. `{"P": "patient", "A": "accession"}`.

Modul: `gdtWriter`

Erzeugt eine GDT-Antwortdatei (Satztyp 6310) mit dem lokalen PDF-Pfad.

Feld	Pflicht	Standard	Beschreibung
`enabled`	nein	`false`	Modul aktivieren.
`templateFile`	nein	`templates/gdt-output.template`	Pfad zum Handlebars-Template (relativ zum Anwendungsverzeichnis oder absolut).
`gdtOutputDir`	ja (wenn enabled)	–	Verzeichnis für die GDT-Antwortdateien. Wird bei Bedarf automatisch angelegt.
`gdtFileName`	ja (wenn enabled)	–	Dateiname der GDT-Antwortdatei. Handlebars-Template, siehe Template-Variablen.
`senderId`	nein	`RadioWeb`	Kennung der sendenden Anwendung in der GDT-Antwortdatei (Feld 8316).

Modul: `hl7Sender`

Sendet eine HL7-MDM^T02-Nachricht per MLLP/TCP an einen HL7-Empfänger.

Eine vollständige Beschreibung des HL7-Nachrichtenformats finden Sie unter HL7 MDM^T02.

Feld	Pflicht	Standard	Beschreibung
`enabled`	nein	`false`	Modul aktivieren.
`templateFile`	ja (wenn enabled)	–	Pfad zum Handlebars-Template für die HL7-Nachricht.
`host`	ja (wenn enabled)	–	Hostname oder IP-Adresse des MLLP-Listeners.
`port`	ja (wenn enabled)	–	TCP-Port des MLLP-Listeners.
queue – Nachrichten-Queue bei Sendefehler			
`queue.enabled`	nein	`false`	Queue aktivieren. Bei Verbindungsfehlern wird die Nachricht in `queue.dir` gespeichert statt verloren zu gehen. Vollständige Beschreibung: HL7-Nachrichten-Queue.
`queue.dir`	ja (wenn queue.enabled)	–	Verzeichnis für zwischengespeicherte HL7-Nachrichten. Wird automatisch angelegt.
`queue.retryIntervalMs`	nein	`60000`	Intervall in Millisekunden, in dem der Daemon die Queue leert. Im CLI-Modus wird die Queue bei jedem Aufruf sofort geleert.

Modul: `hl7Writer`

Schreibt die HL7-MDM^T02-Nachricht als Textdatei in ein lokales Verzeichnis. Nützlich für Tests oder nachgelagerte dateibasierte Verarbeitung. Kann unabhängig von `hl7Sender` aktiviert werden.

Feld	Pflicht	Standard	Beschreibung
`enabled`	nein	`false`	Modul aktivieren.
`templateFile`	ja (wenn enabled)	–	Pfad zum Handlebars-Template für die HL7-Nachricht.
`hl7OutputDir`	ja (wenn enabled)	–	Ausgabeverzeichnis für HL7-Dateien. Wird bei Bedarf automatisch angelegt.
`hl7FileName`	ja (wenn enabled)	–	Dateiname der HL7-Ausgabedatei. Handlebars-Template, siehe Template-Variablen.

Modul: `log`

Schreibt alle Laufzeitausgaben in eine täglich rotierende Logdatei. Die Konsolenausgabe bleibt unverändert aktiv — das Modul ergänzt sie durch eine persistente Protokolldatei.

Feld	Pflicht	Standard	Beschreibung
`enabled`	nein	`false`	Dateiprotokollierung aktivieren.
`level`	nein	`info`	Minimaler Log-Level: `error`, `warn`, `info`, `verbose`, `debug`.
`file`	nein	`logs/gdt-hl7-bridge-%DATE%.log`	Pfad zur Logdatei (relativ zum Anwendungsverzeichnis oder absolut). `%DATE%` wird durch das aktuelle Datum (`YYYY-MM-DD`) ersetzt. Das Verzeichnis wird automatisch angelegt.
`maxSize`	nein	`10m`	Maximale Dateigröße pro Logdatei. Einheiten: `k` (KB), `m` (MB), `g` (GB).
`maxFiles`	nein	`30d`	Aufbewahrungsdauer. Suffix `d` = Tage (z.B. `30d`), ohne Suffix = maximale Dateianzahl (z.B. `14`).

`vars` – Globale Template-Variablen

Das `vars`-Objekt definiert frei belegbare Variablen, die in *allen* Templates (GDT und HL7) verfügbar sind. Typisch sind Praxis- oder Institutionsdaten, die sich selten ändern.

```
"vars": {
  "institute": "Dres. Müller Schmidt",
  "SF2":  "Dres. Müller Schmidt",
  "SF3":  "Albrechtstraße 1",
  "SF6":  "00000 Musterstadt",
  "SF7":  "Germany",
  "SF16": "+49 0815 4711"
}
```

Die Schlüssel können frei gewählt werden und sind im Template direkt als `{{institute}}`, `{{SF2}}` usw. zugreifbar.

Zwei Schlüssel haben im Standard-GDT-Antwort-Template eine besondere Bedeutung:

Schlüssel	Bedeutung
`doc6302`	GDT-Feld 6302 (Befundanforderungskennung / Marker). Wenn leer oder nicht gesetzt, wird das Feld weggelassen.
`doc6304`	GDT-Feld 6304 (Dokumentenname / Titel des Befunds). Wenn leer oder nicht gesetzt, wird das Feld weggelassen.

Feldmapping (`gdtImport.mapping`)

Das `mapping`-Objekt ordnet logischen Bezeichnern (Mapping-Schlüssel) die entsprechenden GDT-Feldkennungen zu. Alle gemappten Felder werden aus der Eingangsdatei ausgelesen und stehen als Template-Variablen zur Verfügung.

Schlüssel	GDT-Feldkennung	Bedeutung
`patid`	`3000`	Patientennummer
`nachname`	`3101`	Nachname des Patienten
`vorname`	`3102`	Vorname des Patienten
`geburtsdatum`	`3103`	Geburtsdatum (GDT-Format TTMMJJJJ)
`accnbr`	`8402`	Auftragsnummer / Accession Number
`fallnummer`	`8402`	Fallnummer (Alias für `accnbr`, selbe Feldkennung)
`studyuid`	`8410`	DICOM Study UID (Belegung ist herstellerabhängig)
`scope`	frei wählbar (optional)	Freigabeebene pro Aufruf, siehe Freigabeebene (`gdtImport.scope`). Nur eintragen, wenn die Praxissoftware ein Feld frei befüllen kann.

NOTE Die GDT-Feldkennungen können angepasst werden, wenn die sendende Praxissoftware abweichende Feldkennungen verwendet.

Pflichtfelder (`gdtImport.required`)

Das `required`-Array enthält die Mapping-Schlüssel, die in der Eingangsdatei zwingend vorhanden sein müssen. Fehlt ein Pflichtfeld, wird die Datei mit einer Fehlermeldung nach `error/` verschoben.

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

Priorität (`gdtImport.priority`)

Das `priority`-Array bestimmt, welcher Wert für den API-Abruf verwendet wird, wenn mehrere gemappte Felder vorhanden sind. Es wird genau der erste Schlüssel verwendet, für den ein Wert in der Eingangsdatei gefunden wurde.

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

Table 1. Beispiele

Vorhandene Felder in der GDT	Verwendeter Parameter	Abfrage
`studyuid` + `patid`	`studyuid`	`?studyuid=2.25.186…`
`accnbr` + `patid`	`accnbr`	`?accnbr=186B7928`
nur `patid`	`patid`	`?patid=3209648408`

Freigabeebene (`gdtImport.scope`)

Beim PDF-Abruf erzeugt das Portal einen Zugangscode. Die Freigabeebene legt fest, welche Studien dieser Zugangscode freigibt.

Wert	Freigabe
`patient`	Alle Studien des Patienten (Patientenfreigabe).
`accession`	Alle Studien der Auftragsnummer. Beim Abruf per `studyuid` sind das alle Studien mit derselben Auftragsnummer wie diese Studie.
`study`	Nur die angefragten Studien (Studienfreigabe): beim Abruf per `studyuid` genau diese Studie, beim Abruf per `accnbr` alle Studien der Auftragsnummer.
`auto`	`accession`, wenn die GDT-Datei eine Auftragsnummer oder Study UID enthält, sonst `patient`.

Die Ebene wird in dieser Reihenfolge ermittelt:

1. GDT-Feld `mapping.scope`, falls es in der Eingangsdatei befüllt ist. Werte werden über `scopeValues` übersetzt, sonst direkt verwendet.
2. `gdtImport.scope` (im Profil oder in der Vorlage, siehe Profile (`profiles`)).
3. Ohne Angabe gilt die Portal-Einstellung **Freigabeebene PDF-Zugangscode** (`token.pdf.max_scope`, Standard `PATIENT`).

Die Portal-Einstellung ist zugleich die weiteste erlaubte Ebene: Ist dort z.B. `ACCESSION` eingestellt, lehnt das Portal Anfragen mit `patient` ab.

In folgenden Fällen wird die Eingangsdatei mit einer `.err`-Datei nach `error/` verschoben, und es wird kein Zugangscode erzeugt:

- ungültiger Wert im GDT-Feld,
- `accession` oder `study` ohne Auftragsnummer und Study UID in der Datei (`SHARE_SCOPE_REQUIRES_STUDY`),
- Ebene weiter als im Portal erlaubt (`SHARE_SCOPE_NOT_ALLOWED`).

Im CLI-Modus wird die Ebene mit `--scope patient|accession|study|auto` angegeben.

Freigabeebene über ein GDT-Feld (z.B. 6200)

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

Profile (`profiles`)

Mit Profilen betreibt eine Bridge-Installation mehrere unabhängige Abläufe, z.B. einen pro Button in der Praxissoftware. Jedes Profil hat ein eigenes Eingangsverzeichnis und kann alle Module abweichend konfigurieren, etwa die Freigabeebene, einen anderen HL7-Port oder einen zusätzlichen `hl7Writer`.

- **Ohne** `profiles` ist die gesamte Konfiguration das einzige Profil (`default`). Bestehende Konfigurationen funktionieren unverändert.
- **Mit** `profiles` ist alles außerhalb von `profiles` nur noch Vorlage und läuft selbst nicht mit. Es laufen ausschließlich die Profile.
- Ein Profil enthält nur die Abweichungen von der Vorlage:
  - Objekte werden zusammengeführt, d.h. `"hl7Sender": { "port": 2334 }` ändert nur den Port.
  - Arrays und Einzelwerte ersetzen den Vorlagenwert (z.B. `priority`).
  - `null` entfernt einen Vorlagenwert.
- `"enabled": false` direkt im Profil schaltet das ganze Profil ab.
- `log` gilt global und darf in Profilen nicht vorkommen. Logzeilen enthalten den Profilnamen, z.B. `Processing [accession]: …`.
- `portalUrl` und `apiToken` können pro Profil abweichen, z.B. für eine zweite Praxis mit eigenem Integrations-Token.
- Schlüssel, die mit `_` beginnen, gelten als Kommentar und werden ignoriert.

```
{
  "portalUrl": "https://portal.ihre-praxis.de",
  "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 }
    }
  }
}
```

Im Beispiel erzeugt `patient` eine Patientenfreigabe und eine GDT-Antwort. `accession` gibt nur die Auftragsnummer frei, sendet zusätzlich HL7 an Port 2334 und legt die HL7-Nachricht auch als Datei ab.

Regeln, die beim Start geprüft werden

- Alle Pflichtfelder werden pro Profil geprüft (nach dem Zusammenführen). Fehlermeldungen nennen das Profil, z.B. `"profiles.accession.hl7Sender.host" missing`.
- Jedes Profil braucht ein eigenes `gdtImport.gdtInputDir`.
- Profile, deren HL7-Nachrichten an verschiedene Empfänger (`host:port`) gehen, brauchen jeweils ein eigenes `hl7Sender.queue.dir`. Die Queue eines Verzeichnisses wird immer an genau einen Empfänger zugestellt.

Im CLI-Modus wählen Sie das Profil mit `--profile <name>`. Gibt es nur ein Profil, ist die Angabe nicht nötig.

TIP Mit `gdt-hl7-bridge --print-profiles` prüfen Sie nach jeder Änderung, was jedes Profil tatsächlich tut. Die Ausgabe zeigt die zusammengeführten Werte und markiert mit `*`, was aus dem Profil stammt.

Template-Variablen

Die folgenden Variablen stehen in den Dateinamen-Templates (`pdfFileName`, `gdtFileName`, `hl7FileName`) zur Verfügung. Alle im `mapping` definierten Schlüssel stehen ebenfalls zur Verfügung, sofern das entsprechende Feld in der Eingangsdatei vorhanden ist.

Variable	Inhalt
`{{patid}}`	Patientennummer (GDT-Feld 3000)
`{{nachname}}`	Nachname (GDT-Feld 3101)
`{{vorname}}`	Vorname (GDT-Feld 3102)
`{{geburtsdatum}}`	Geburtsdatum im Format TTMMJJJJ (GDT-Feld 3103)
`{{accnbr}}`	Auftragsnummer / Accession Number (GDT-Feld 8402)
`{{fallnummer}}`	Fallnummer – identisch mit `accnbr` (GDT-Feld 8402)
`{{studyuid}}`	DICOM Study UID (GDT-Feld 8410, herstellerabhängig)
`{{scope}}`	Ermittelte Freigabeebene (`patient`, `accession`, `study`; leer, wenn die Portal-Einstellung gilt)
`{{timestamp}}`	Unix-Zeitstempel in Millisekunden
`{{date}}`	Aktuelles Datum im Format `JJJJ-MM-TT`
`{{time}}`	Aktuelle Uhrzeit im Format `HH-MM-SS`
`{{inputFile}}`	Dateiname der Eingangsdatei ohne Dateiendung
`{{<key>}}`	Jede Variable aus `vars` steht direkt unter ihrem Schlüsselnamen zur Verfügung

GDT-Antwort-Template

Das Template `templates\gdt-output.template` steuert den Aufbau der erzeugten GDT-Antwortdatei im Format GDT 2.1. Es verwendet Handlebars-Syntax.

Der Helper `{{gdtLine "FFFF" "Inhalt"}}` erzeugt eine korrekt formatierte GDT-Zeile inklusive Längenprefix und CRLF.

Standard-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-Kontext

Variable	Inhalt
`fields`	Alle GDT-Felder der Eingangsdatei, zugreifbar per Feldkennung: `(lookup fields "3000")`
`pdfPath`	Absoluter lokaler Pfad zur heruntergeladenen PDF-Datei (leer, wenn `pdfFile.enabled: false`)
`senderId`	Wert aus `gdtWriter.senderId`
`doc6302`	Wert aus `vars.doc6302` – optionaler Befundanforderungsmarker (Feld 6302)
`doc6304`	Wert aus `vars.doc6304` – optionaler Dokumentenname (Feld 6304)
`{{<key>}}`	Alle weiteren Variablen aus `vars`

Erzeugte GDT-Felder

Feldkennung	Bedeutung
`8000`	Satztyp: `6310` (Daten einer Untersuchung)
`9206`	GDT-Version: `2.10`
`8316`	ID des Senders (`senderId`)
`8315`	ID des Empfängers – aus Eingabefeld 8316 (Kennung der einsendenden Praxissoftware), nur wenn vorhanden
`8310`	Auftragsnummer des Einsenders – aus Eingabefeld 8402 (Accession Number), nur wenn vorhanden
`3000` / `3101` / `3102` / `3103` / `3104`	Patientendaten aus der Eingangsdatei (nur wenn vorhanden)
`6302`	Befundanforderungsmarker (aus `vars.doc6302`, optional)
`6303`	Dateiformat: `PDF` (immer gesetzt, wenn `pdfPath` vorhanden)
`6304`	Dokumentenname (aus `vars.doc6304`, optional)
`6305`	Verweis auf die Datei – lokaler Pfad zur PDF-Datei