

GMC Systems mbH

SCHNITTSTELLENBESCHREIBUNG „KOMMANDODATEI GMC PADOK“

Version: 0.60
Datum: 11.06.2012

vorgelegt von
GMC System mbH
Albert-Einstein-Str. 3
98693 Ilmenau

Ansprechpartner: Andreas Heyn (email: ahe@gmc-systems.de)

INHALTSVERZEICHNIS

| | | |
|---------|--|----|
| 1 | Dokumenteninformation..... | 3 |
| 1.1 | <i>Status</i> | 3 |
| 1.2 | <i>Revisionsliste</i> | 3 |
| 2 | Einleitung | 4 |
| 3 | Aufbau der Kommandodatei | 5 |
| 3.1 | <i>Allgemein</i> | 5 |
| 3.2 | <i>Das Root-Element mcc_cmd_file</i> | 6 |
| 3.2.1 | <i>Das Element user</i> | 7 |
| 3.2.2 | <i>Das Element sync_event</i> | 8 |
| 3.2.3 | <i>Das Element result_file</i> | 9 |
| 3.3 | <i>Kommandoliste</i> | 10 |
| 3.3.1 | <i>Allgemeiner Kommandoaufbau</i> | 10 |
| 3.3.2 | <i>Das Containerelement command_list</i> | 10 |
| 3.4 | <i>Das Kommando CREATE_MAIL (Beispiel)</i> | 11 |
| 3.4.1 | <i>Das Element receiver</i> | 13 |
| 3.4.2 | <i>Das Element patient</i> | 13 |
| 3.4.2.1 | <i>Das Element patient.patient_id</i> | 14 |
| 3.4.2.2 | <i>Das Element patient.import_file</i> | 14 |
| 3.4.3 | <i>Das Element main_document</i> | 16 |
| 3.4.4 | <i>Das Element attachment</i> | 19 |
| 3.4.5 | <i>Das Element exec_mode</i> | 20 |
| 3.4.5.1 | <i>Kommandoabarbeitung im „HIDDEN“-Modus</i> | 21 |
| 4 | Anhang..... | 22 |
| 4.1 | <i>Tabellenverzeichnis</i> | 22 |
| 4.2 | <i>Abbildungsverzeichnis</i> | 22 |
| 4.3 | <i>XML-Code-Verzeichnis</i> | 22 |

1 Dokumenteninformation

1.1 Status

Entwurf

1.2 Revisionsliste

| Version | Autor | Inhalt | Datum |
|----------------|--------------|---|--------------|
| 0.5 | AHE, UWO | Erster Entwurf der Schnittstellenbeschreibung | 29.02.2012 |
| 0.6 | AHE | Korrekturen, Ergänzungen | 11.06.2012 |

2 Einleitung

Neben der allgemeinen Kommandozeilenschnittstelle erlaubt GMC PaDok die Steuerung über eine Kommandodatei im XML-Format.

Über den Kommandozeilenparameter **/cmdFile:Dateipfad** kann der absolute Dateipfad zu der zu interpretierenden Kommandodatei angegeben werden. Alle weiteren Kommandozeilenparameter werden in diesem Fall ignoriert.

Beispiel: `padok.exe /cmdFile:"c:\cmd.xml"`

Vorteile der Kommandodatei gegenüber der allg. Kommandozeilenschnittstelle:

- Komplexe Kommandos über allg. Kommandozeilenparameter nicht mehr realisierbar
- Übergabe und Abarbeitung von Kommandolisten mit einem einzelnen Programmaufruf
- Strukturierter Aufbau; genaue Definition der einzelnen Kommandos und der zugehörigen Parameter
- Keine Passwörter und PINs in Klartext auf Kommandozeile

Im Folgenden wird der Aufbau der GMC PaDok Kommandodatei erläutert.

Zur Darstellung über welche Klasse und über welches Attribut ein Datenfeld abgebildet wird, wird folgende Kurzbeschreibung verwendet.

Element..... **Datenfeld**
Datentyp [n..m]

| | |
|------------|--|
| Element: | Name des XML-Datenelements |
| Datenfeld: | Name des Datenfeldes, dessen Darstellung beschrieben wird. |
| Datentyp: | Gibt den Datentyp des Elements an. |
| [n..m]: | Kardinalität des Datenfeldes |

3 Aufbau der Kommandodatei

3.1 Allgemein

Für eine detaillierte Beschreibung der Dokumentenstruktur sei auch auf die zugehörige Schemadatei [GMCPaDok.Command.xsd](#) verwiesen.

XML-Namespaces werden in der Kommandodatei nicht verwendet. Zulässige Zeichensätze sind ISO-8859-1 und UTF-8.

GMC PaDok Kommandodateien beginnen mit dem Wurzelement *mcc_md_file*, der grobe Aufbau ist im folgenden Übersichtsbeispiel gegeben.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mcc_cmd_file
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.0"
  synchron_mode="true" >

  <!-- Header -->
  ...

  <!-- Kommandoliste -->
  <command_list>
  ...
  </command_list>
</mcc_cmd_file>
```

XML-Code 1 Aufbau der Kommandodatei

3.2 Das Root-Element *mcc_cmd_file*

Über das Root-Element *mcc_cmd_file* können neben der eigentlichen abzuarbeitenden Kommandoliste (siehe Kapitel 3.3) weitere, für die Befehlsabarbeitung benötigte Daten strukturiert übergeben werden.

- Für das *@version* Attribut ist der Wert „1.0“ fest vorgegeben.
- Das Attribut *@synchron_mode* wird derzeit noch ignoriert.
- Das Attribut *@halt_on_error* legt fest, ob die Kommandoabarbeitung im Fehlerfall abgebrochen werden soll. Tritt bei der Abarbeitung eines Kommandos ein Fehler auf, werden nachfolgende Kommandos nicht mehr abgearbeitet (default=true).

Im Folgenden werden alle relevanten Header-Elemente, deren Datentyp bzw. Bedeutung und deren Kardinalität aufgeführt. Dazu werden XML-Fragmente als Beispiele gezeigt.

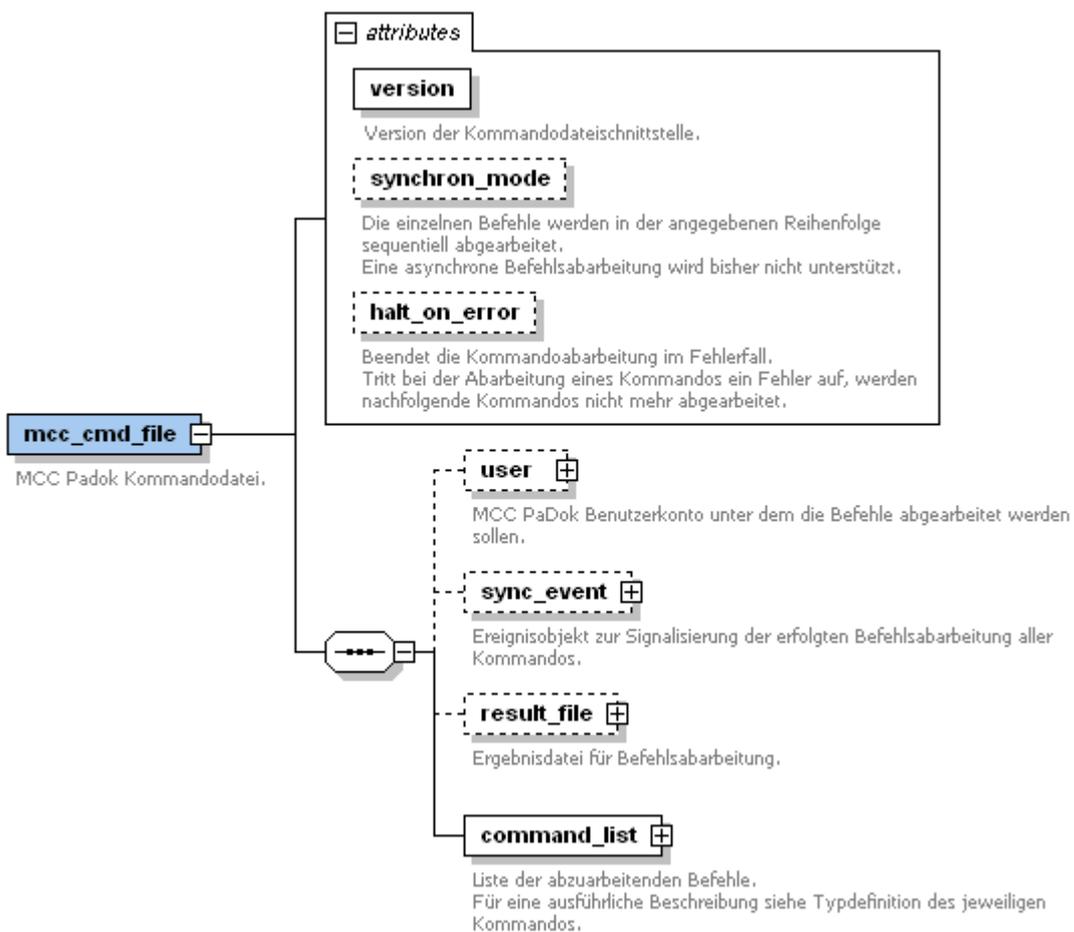


Abbildung 1 Grundstruktur der Kommandodatei (*mcc_cmd_file*)

3.2.1 Das Element *user*

userMCC PaDok Benutzerkonto
mcc_user_account_type [0..1]

Das Element *user* legt das GMC PaDok Benutzerkonto fest, unter dem die Befehle abgearbeitet werden sollen.

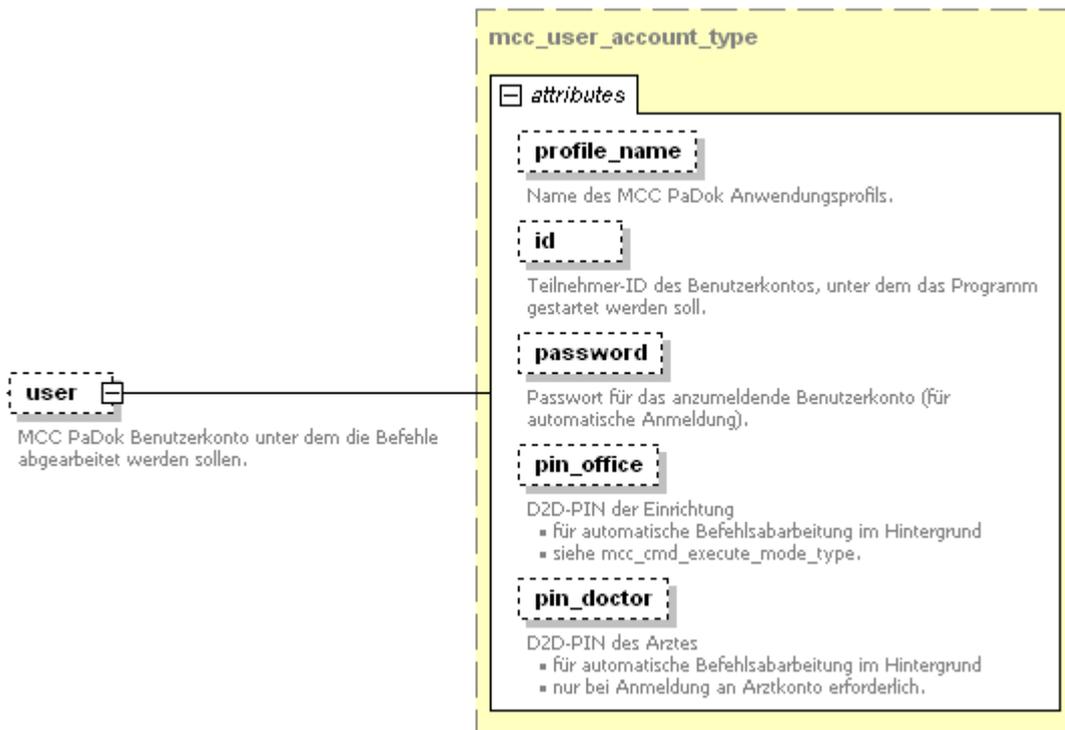


Abbildung 2 `mcc_cmd_file.user`

```
<user profile_name="Aufnahme" id="1999432" password="geheim" pin_office="123456" pin_doctor="234567" />
```

XML-Code 2 `mcc_cmd_file.user`

- Das Attribut `@profile_name` legt das GMC PaDok Anwendungsprofil fest, unter dem das Programm gestartet werden soll.
- Im `@id` Attribut wird die Teilnehmer-ID des Benutzerkontos angegeben, unter dem die Befehle abgearbeitet werden sollen.
- Das Attribut `@password` erhält das Passwort für das anzumeldende Benutzerkonto (für automatische Anmeldung).
- Über das Attribut `@pin_office` kann die D2D-PIN der Einrichtung angegeben werden (nur erforderlich bei automatischer Befehlsabarbeitung im Hintergrund).
- Über das Attribut `@pin_doctor` kann die D2D-PIN des Arztkontos angegeben werden (nur erforderlich bei Anmeldung an einem Arztkonto und bei automatischer Befehlsabarbeitung im Hintergrund).

Erläuterung GMC PaDok Anwendungsprofile

Für eine GMC PaDok Installation können mehrere Anwendungsprofile verwaltet werden. Alle Programmeinstellungen, einschließlich der Pfade zum Datenverzeichnis und zum D2D-Client werden pro Anwendungsprofil separat gespeichert. Somit können unterschiedliche Anwendungsfälle (z.B. mehrere Praxen / Abteilungen) für eine GMC PaDok-Installation konfiguriert werden (siehe [Konfiguration der GMC PaDok -Installation](#)).

3.2.2 Das Element `sync_event`

**`sync_event`Synchronisationsobjekt
`mcc_sync_event_type [0..1]`**

Angabe des Namens eines durch den aufrufenden Prozess erzeugten Synchronisationsobjekts zur Signalisierung der erfolgten Abarbeitung der Kommandoliste.

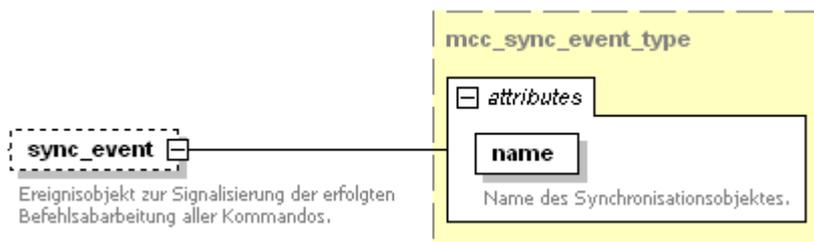


Abbildung 3 `mcc_cmd_file.sync_event`

```
<sync_event name="Name des Synchronisationsobjektes" />
```

XML-Code 3 `mcc_cmd_file.sync_event`

Erläuterung

An den Prozess GMC PaDok (`padok.exe`) übergebene Befehle werden aus Sicht des Aufrufers asynchron ausgeführt. Soll auf die Abarbeitung der Befehle gewartet werden, kann dem Element `sync_event` der Name eines Synchronisationsobjektes übergeben werden.

Nach erfolgter Befehlsabarbeitung wird das Synchronisationsobjekt in den signalisierten Zustand versetzt.

3.2.3 Das Element *result_file*

result_file.....Ergebnisdatei
mcc_result_file_type [0..1]

Angabe des absoluten Dateipfades zu einer zu erstellenden Ergebnisdatei.

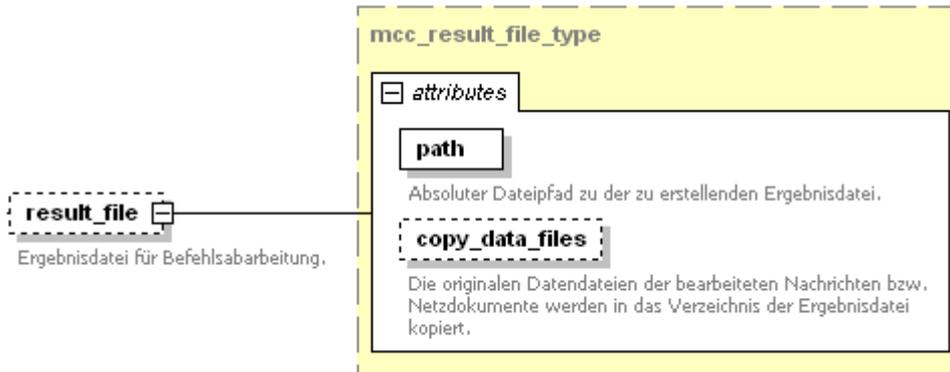


Abbildung 4 `mcc_cmd_file.result_file`

```
<result_file path="c:\test\mcc_result.xml" copy_data_files="false" />
```

XML-Code 4 `mcc_cmd_file.result_file`

- Im `@path` Attribut wird der absolute Dateipfad zu der zu erstellenden Ergebnisdatei angegeben. Das Zielverzeichnis muss bereits existieren und Schreibrechte vorhanden sein.
- Wird das Attribut `@copy_data_files` auf `true` gesetzt, werden zusätzlich zur Ergebnisdatei die originalen Datendateien der bearbeiteten Nachrichten bzw. Netzdokumente in das Verzeichnis der Ergebnisdatei kopiert.

Erläuterung

Nach Abarbeitung der Kommandoliste wird die Ergebnisdatei von GMC PaDok unter dem angegebenen Dateipfad erstellt. Die Ergebnisdatei enthält Listen aller während der Befehlsabarbeitung erstellten, bearbeiteten, empfangenen und versendeten Nachrichten bzw. Netzdokumente.

Die Ergebnisdatei besitzt den folgenden Aufbau: [GMCPaDok.CommandResult.xsd](#)

Wurde über das optionale Element `sync_event` der Name eines Ereignisobjekts angegeben, so wird das Ereignisobjekt in den signalisierten Zustand versetzt (siehe Kapitel 3.2.2). Andernfalls wird im gleichen Verzeichnis eine leere Semaphor-Datei mit gleichem Namen und der Dateiendung `.sem` erstellt (z.B. `c:\test\mcc_result.sem`).

Das aufrufende Programm detektiert entweder die Statusänderung des Ereignisobjekts oder das Erstellen der Semaphor-Datei. Der Aufrufer ist für das Löschen evtl. erstellter Dateien verantwortlich.

3.3 Kommandoliste

3.3.1 Allgemeiner Kommandoaufbau

Die Liste der abzuarbeitenden Kommandos setzt sich aus einem oder mehreren Kommandoobjekten zusammen. Die gemeinsamen Eigenschaften von Kommandoobjekten werden durch die abstrakte Basisklasse *mcc_cmd_base_type* definiert.



Abbildung 5 Allgemeiner Kommandoaufbau (*mcc_cmd_base_type*)

- Im *@command_id* Attribut kann eine eindeutige Kommando-ID zur späteren Identifizierung des Kommandos bzw. zur Referenzierung des Kommandos in der Ergebnisdatei übergeben werden. Wird keine Kommando-ID angegeben, so wird der eins-basierte Index des Kommandos innerhalb der Kommandoliste verwendet.
- Dem *@delay* Attribut kann eine Verzögerung in Millisekunden für den Start der Kommandoabarbeitung übergeben werden.

3.3.2 Das Containerelement *command_list*

Das Containerelement *command_list* enthält eine Liste aller abzuarbeitenden Kommandos bei diesem Programmaufruf. Eine detaillierte Beschreibung der einzelnen Kommandos finden Sie in der Schemadatei [GMCPaDok.Command.xsd](#).

Die folgende Abbildung zeigt alle derzeit unterstützten Kommandotypen.

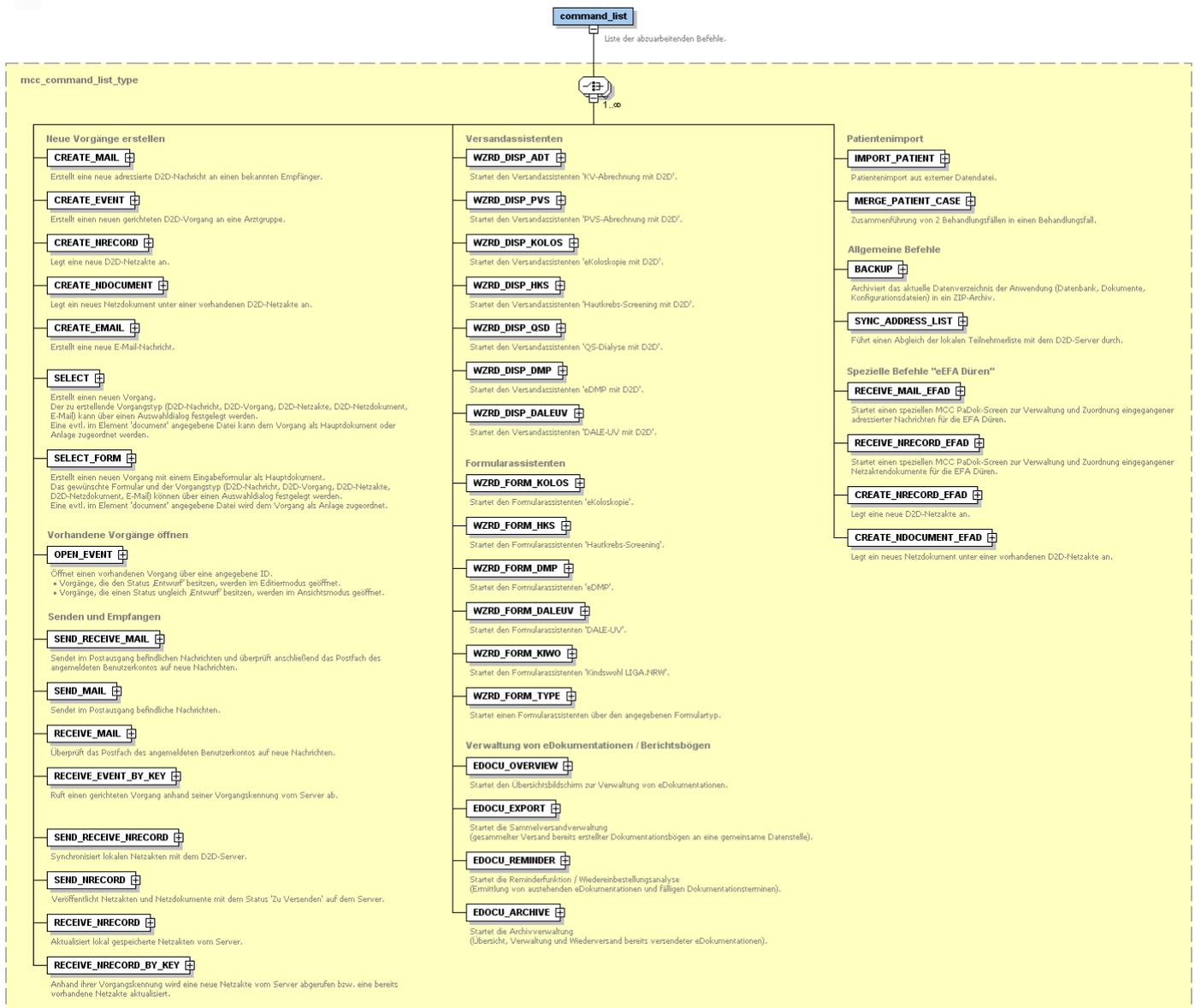


Abbildung 6 Bisher unterstützte GMC PaDok Befehle (command_list)

Im Folgenden wird exemplarisch das Kommando *CREATE_MAIL* (Erstellung einer neuen adressierten D2D-Nachricht an einen bekannten Empfänger) beschrieben.

3.4 Das Kommando *CREATE_MAIL* (Beispiel)

Über das Element *CREATE_MAIL* kann eine neue adressierte D2D-Nachricht an einen bekannten Empfänger erstellt und versendet werden.

Der Aufbau einer D2D-Nachricht ist durch das D2D-Versandverfahren vorgegeben. Eine Nachricht besteht immer aus einem Absender, einem Empfänger, einem Hautdokument und einer beliebigen Anzahl von Anlagen. Der Absender wird durch das angemeldete GMC PaDok Benutzerkonto festgelegt (siehe Kapitel 3.2.1) und muss daher an dieser Stelle nicht angegeben werden.

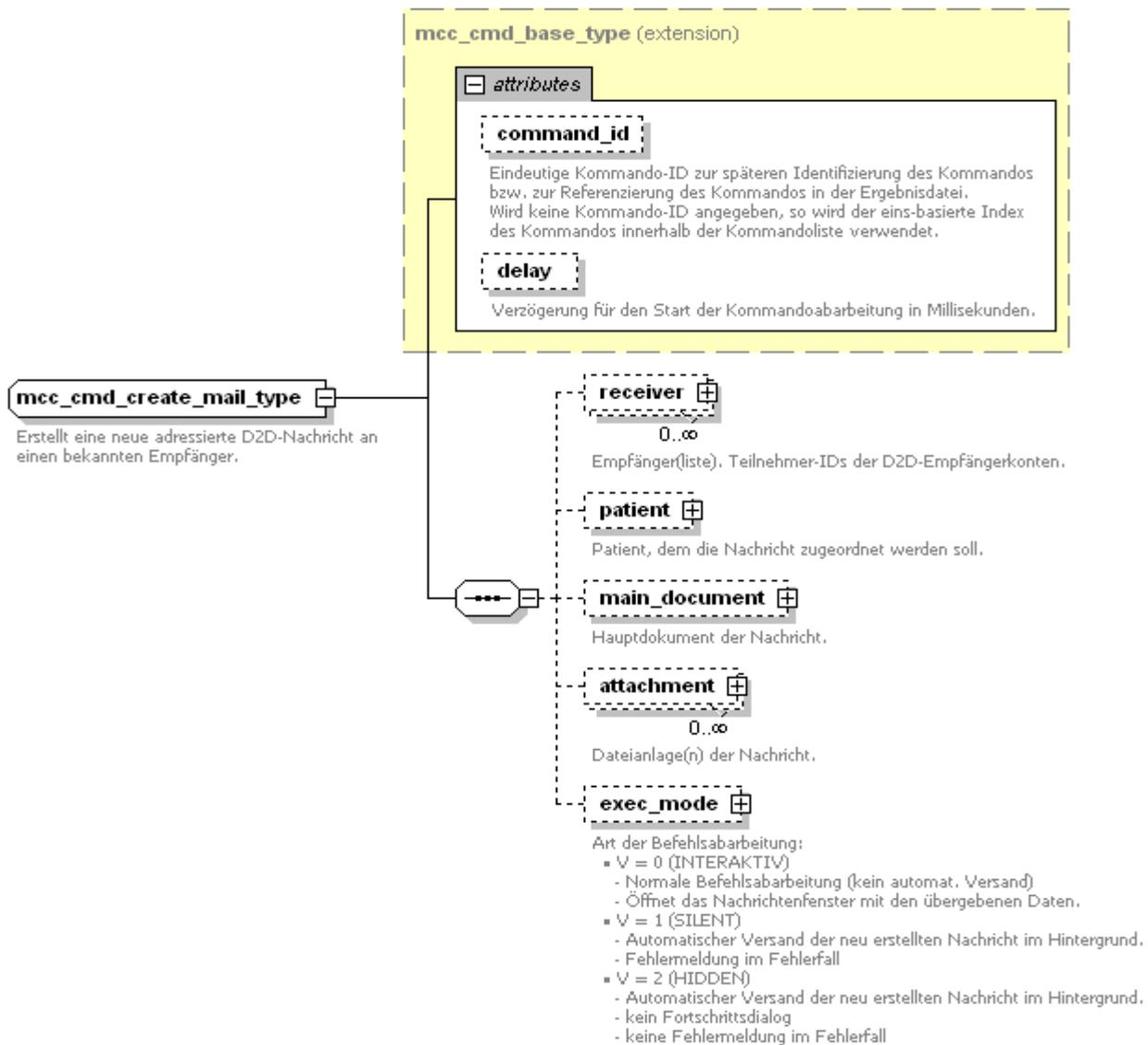


Abbildung 7 Das Kommando CREATE_MAIL

Der Coderahmen für das Element CREATE_MAIL sieht wie folgt aus:

```

<create_mail command_id="1" delay="0" halt_on_error="true" >
  <!-- Empfänger -->
  <receiver id="1999432" />

  <!-- Patient -->
  <patient> ... </patient>

  <!-- Hauptdokument -->
  <main_document path="c:\docs\test.pdf" comment="Testmail" file_type="1017" file_format="8" />

  <!-- Anlage 1 -->
  <attachment path="c:\docs\test.jpg" comment="Testanlage" file_type="5002" file_format="22" />
  ...

  <!-- Automatischer Versand im Hintergrund -->
  <exec_mode V="1" />
</mcc_cmd_file>

```

XML-Code 5 Das Kommando CREATE_MAIL

Nachfolgend werden die Kindelemente des Kommandos `CREATE_MAIL` aufgeführt.

3.4.1 Das Element *receiver*

receiver *Empfänger*
mcc_receiver_d2d_type [0..*]

Die Empfänger werden über die Teilnehmer-ID ihrer D2D-Empfängerkonten festgelegt.

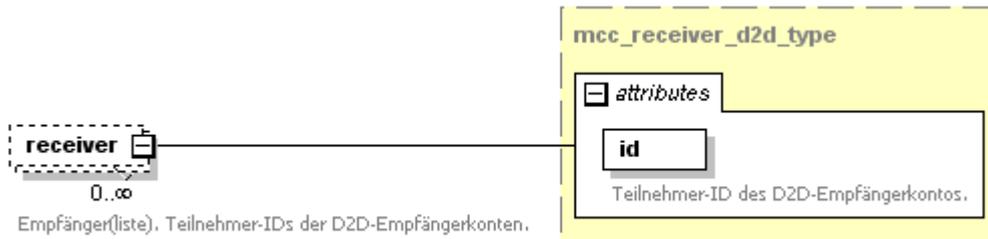


Abbildung 8 `CREATE_MAIL.receiver`

```
<receiver id="1909432" />
```

XML-Code 6 `CREATE_MAIL.receiver`

3.4.2 Das Element *patient*

patient *Patient*
mcc_patient_type [0..1]

Optional kann der Patient angegeben werden, dem die Nachricht zugeordnet werden soll. Die Angabe des Patienten kann dabei über eine externe Patienten-ID (PVS-Nummer, KIS-Nummer) oder über eine zu importierende Patientendatendatei erfolgen.

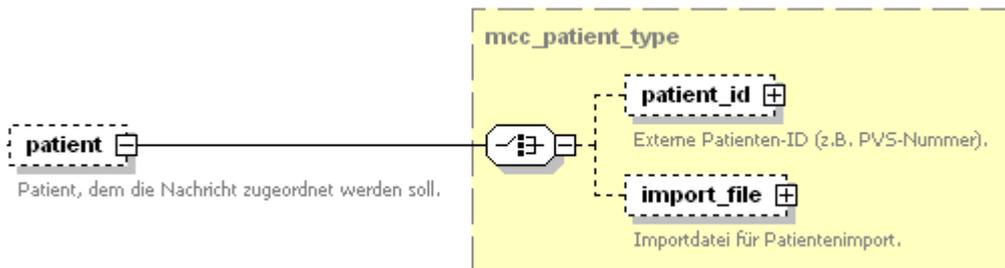


Abbildung 9 `CREATE_MAIL.patient`

3.4.2.1 Das Element `patient.patient_id`

Wird der Patient über eine externe Patienten-ID angegeben, muss der Patient bereits in GMC PaDok existieren.

Das Kindelement `patient_id` besteht aus den Attributen `@V` und `@visit_no`.

- Im Attribut `@V` wird die numerische Patienten-ID eingetragen.
- Dem optionalen Attribut `@visit_no` kann die Besuchs- bzw. Fallnummer übergeben werden, der die Nachricht zugeordnet werden soll.

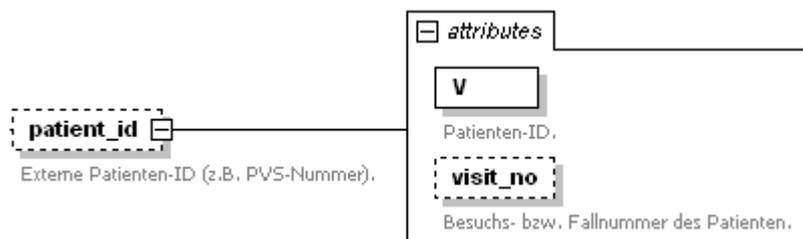


Abbildung 10 `CREATE_MAIL.patient.patient_id`

Als Beispiel für die Angabe über eine externe Patienten-ID sei hier folgender Code angegeben.

```
<patient>
  <patient_id V="1042" visit_no="133345742"/>
</patient>
```

XML-Code 7 `CREATE_MAIL.patient.patient_id`

3.4.2.2 Das Element `patient.import_file`

Bei Angabe einer zu importierende Patientendatei werden die Stamm- und Versichertendaten in das System importiert. Nicht vorhandene Patienten werden neu angelegt, vorhandene Patienten werden aktualisiert. Die Nachricht wird dem zuletzt importierten Patienten (bzw. dem zuletzt importierten Behandlungsfall) zugeordnet. Das Kindelement `import_file` besteht aus den Attributen `@path`, `@type` und `@auto_mode`.

- Im Attribut `@path` wird der absolute Dateipfad zur Importdatei angegeben.
- Über das Attribut `@type` wird der Typ der Importdatei angegeben. Mögliche Werte für das `@type`-Attribut sind in Tabelle 1 aufgeführt.
- Das Attribut `@auto_mode` aktiviert den automatischer Dateiimport (minimale Nutzerinteraktion, keine Anzeige der importierten Daten in der Patientenmaske).

| Code | Definition |
|------|--|
| BDT | BDT- / GDT Import |
| HL7 | HL7v2 / HL7v2.xml Import |
| CDA | CDA Import (einschl. Sciphox Dokumente) |
| CDA2 | CDA Release 2 Import (einschl VHitG) |
| KVK | spezieller KVK- / eGK-Datenimport (internes Dateiformat) |

Tabelle 1 Das `type`-Attribut im Element `import_file`

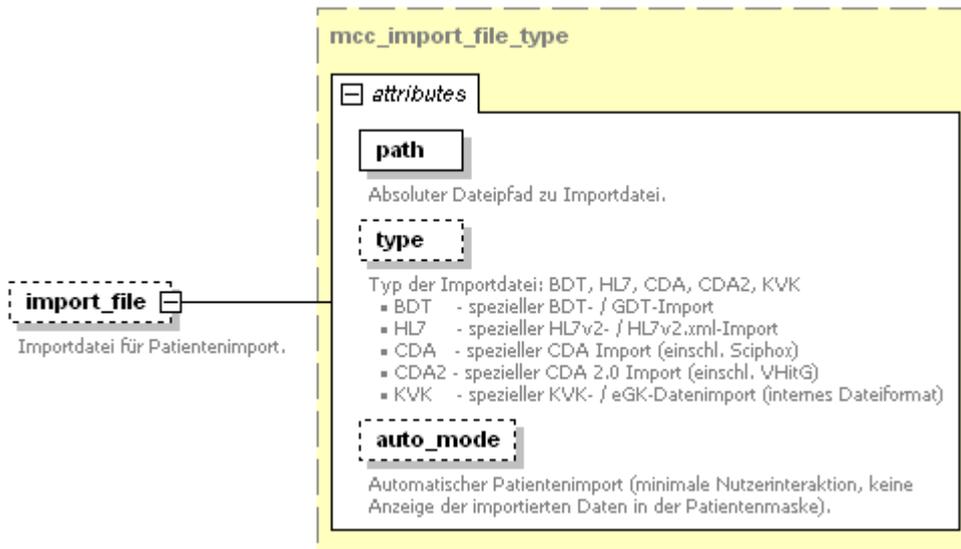


Abbildung 11 CREATE_MAIL.patient.import_file

Als Beispiel für die Angabe einer Importdatei sei hier der folgende Code angegeben.

```
<patient>
  <import_file path="c:\pat.bdt" type="BDT" auto_mode="true"/>
</patient>
```

XML-Code 8 CREATE_MAIL.patient.import_file

3.4.3 Das Element *main_document*

main_document Hauptdokument der Nachricht
mcc_document_type [0..1]

Das Hauptdokument einer D2D-Nachricht wird über das Element *main_document* beschrieben.

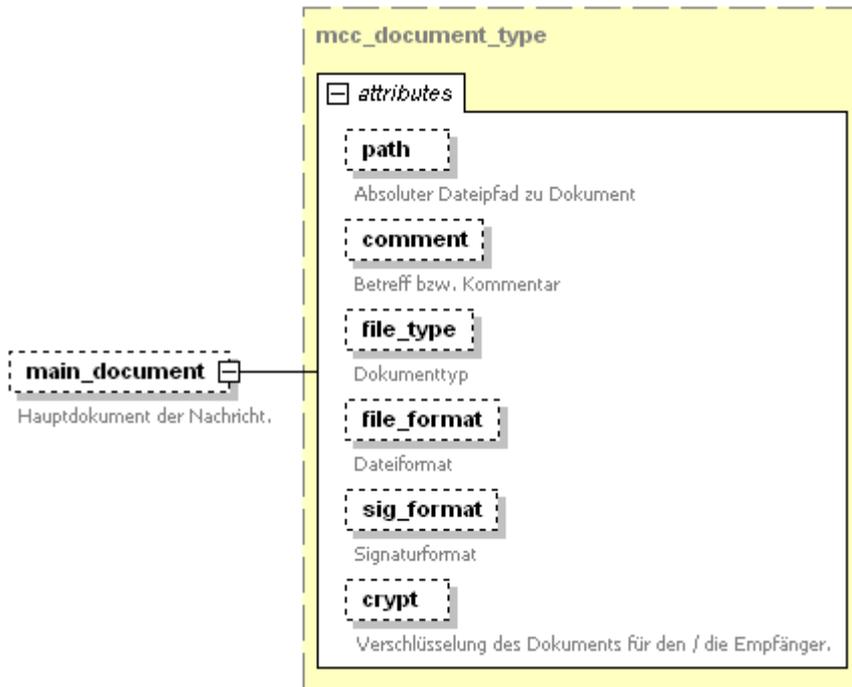


Abbildung 12 CREATE_MAIL.maindocument

- Im Attribut *@path* ist der absolute Dateipfad zum Dokument anzugeben.
- Das Attribut *@comment* hält den Kommentar der Nachricht.
- Über das Attribut *@file_type* wird der Dokumententyp festgelegt. Bisher definierte Dokumententypen sind in Tabelle 2 aufgeführt. Die Definition weiterer, benutzerdefinierter Dokumententypen ist möglich. Wird das Attribut *@file_type* nicht angegeben, so wird der Wert 1017 (Arztbrief) verwendet.
- Über das Attribut *@file_format* kann das Dateiformat des Hauptdokuments übergeben werden. Bisher definierte Dateiformate sind in Tabelle 3 aufgeführt. Die Definition weiterer, benutzerdefinierter Dateiformate ist möglich. Wird das Attribut *@file_format* nicht angegeben, so wird versucht, das Dateiformat von der Dateiendung des übergebenen Dokuments abzuleiten.
- Das Attribut *@sig_format* legt das gewünschte Signaturformat für das übergebene Dokument fest. Die derzeit unterstützten Signaturformate werden in Tabelle 4 aufgelistet.
- Das Attribut *@crypt* legt fest, ob das übergebene Dokument für den/die Empfänger verschlüsselt werden soll (wird derzeit noch ignoriert).

Der Coderahmen für das Element *main_document* sieht wie folgt aus:

```
<main_document path="c:\docs\test.pdf" comment="Testmail" file_type="1017" file_format="8" sig_format="3"/>
```

XML-Code 9 CREATE_MAIL.main_document

| Code | Definition |
|-------------|--|
| 1001 | Krankschreibung, AU |
| 1002 | eEinweisung |
| 1006 | eÜberweisung |
| 1012 | Häusliche Pflege |
| 1016 | eRezept |
| 1019 | Notfalldaten (VHiTG) |
| | |
| 1017 | eArztbrief (allgemein, Sciphox) |
| 1018 | eArztbrief (VHiTG + PDF) |
| 1025 | eArztbrief (VHiTG + PDF + QES) |
| 1026 | PDF-Dokument zu eArztbrief |
| 1027 | XML-Dokument zu eArztbrief |
| | |
| 1030 | eNachricht |
| | |
| 5000 | CT |
| 5001 | Ultraschall |
| 5002 | Röntgenbild |
| 5003 | EKG |
| 5004 | EEG |
| | |
| 6002 | KV-Abrechnung |
| 6022 | KV-Abrechnung mit signierter Sammelerklärung |
| 6023 | Separate signierte Sammelerklärung zur KV-Abrechnung |
| 6024 | KVDT-Begleitdatei |
| 6025 | KVDT-Sammelerklärung (PDF) |
| 6026 | KVDT-Sammelerklärung (XML) |
| | |
| 6001 | eKoloskopie-Bericht |
| 6007 | Hautkrebs-Screening-Bericht |
| 6008 | Qualitätssicherung Dialyse (DIA-Archiv) |
| 6009 | Qualitätssicherung Dialyse (BERST-Archiv) |
| 6010 | abD-Begleitdatei |
| | |
| 6004 | ePVS-Abrechnung |
| | |
| 6011 | eDMP-Bericht |
| 6012 | eDMP-Begleitdatei |
| | |
| 6003 | DALE-UV-Bericht |
| 6015 | DALE-UV-Auftragsdatei |
| | |
| 6016 | EDA-UK-Bericht (Einsendung) |
| 6017 | EDA-UK-Bericht (Rücksendung) |
| 6018 | EDA-UK-Auftragsdatei |
| 6019 | EDA-UK-Anhangscontainer |
| | |
| 6040 | U-Teilnahmebescheinigung LZG.NRW |
| 6041 | Meldepflichtige Infektionskrankheiten (IfSG) |
| | |
| 9999 | Sonstige |
| | |
| 10001 | Qmax Praxisnetz |
| | |
| 20001 | Fluktuationsbogen KfH |
| 20002 | Interne Kommunikation KfH |

| | |
|-------|---|
| | |
| 30000 | Empfangsbestätigung eNachricht, allgemein |
| 30001 | Empfangsbestätigung eArztbrief |
| 30002 | Empfangsbestätigung KV-Abrechnung |
| | |
| 30101 | (Fachliche) Rückmeldung KV-Abrechnung |
| 30102 | Quittung eDMP |
| 30103 | Quittung DALE-UV |
| 30104 | Quittung EDA-UK |
| 30105 | Quittung ePVS-Abrechnung |
| | |
| 30110 | Quittung abD |
| 30111 | Quittung eKoloskopie |
| 30112 | Quittung Hautkrebs-Screening |
| 30113 | Quittung QS-Dialyse (DIA-Archiv) |
| 30114 | Quittung QS-Dialyse (BERST-Archiv) |

Tabelle 2 Dokumententypen

| Code | Definition |
|------|---|
| 0 | BDT |
| 1 | LDT |
| 2 | XML |
| 3 | TIFF |
| 4 | RTF |
| 5 | MS Office |
| 6 | GIF |
| 7 | XAF |
| 8 | PDF |
| 9 | HL7 |
| 10 | ZIP |
| 11 | HTML |
| 12 | KVDT (KVDT-Abrechnungsdatei) |
| 13 | XKM (verschlüsselt mit KBV-Kryptomodul) |
| 21 | TXT (Plain Text) |
| 22 | JPG |
| 23 | BMP |
| 24 | PNG |
| 25 | EML |
| 99 | Sonstige |

Tabelle 3 Dateiformate

| Code | Definition |
|------|--|
| 0 | UNSIGNED (unsignierter Versand) |
| 1 | PRESIGNED (Externe Signatur; das Dokument wird GMC PaDok schon signiert übergeben) |
| 2 | PKCS#7 |
| 3 | PDF |
| 4 | XML-DSig |

Tabelle 4 Signaturformate

Bei Angabe der Signaturformate 2,3 bzw. 4 erfolgt die Signatur des übergebenen Dokumentes durch GMC PaDok (qualifizierte Signatur über Signaturkarte). Werden der Nachricht mehrere zu signierende Dokumente übergeben, so wird eine Stapelsignatur durchgeführt (falls von der Signaturkarte unterstützt).

3.4.4 Das Element *attachment*

attachment.....Anlagen der Nachricht
mcc_document_type [0..]*

Die Anlagen der D2D-Nachricht können über das Element *attachment* hinzugefügt werden. Die Struktur des Typs *mcc_document_type* wurde bereits ausführlich im Kapitel 3.4.3 erläutert.

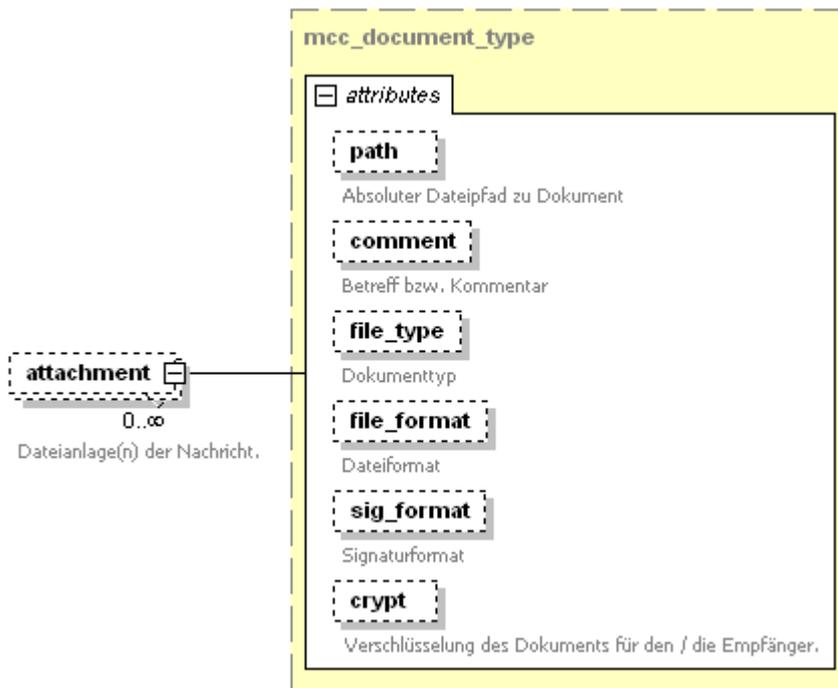


Abbildung 13 CREATE_MAIL.attachment

3.4.5 Das Element exec_mode

exec_mode**Art der Befehlsabarbeitung**
mcc_cmd_execute_send_mode_type [0..1]

Die Art der Befehlsabarbeitung kann über das Element *exec_mode* festgelegt werden. Für das Kommando *CREATE_MAIL* steuert *exec_mode* den automatischen Versand der neu erstellten D2D-Nachricht im Hintergrund.

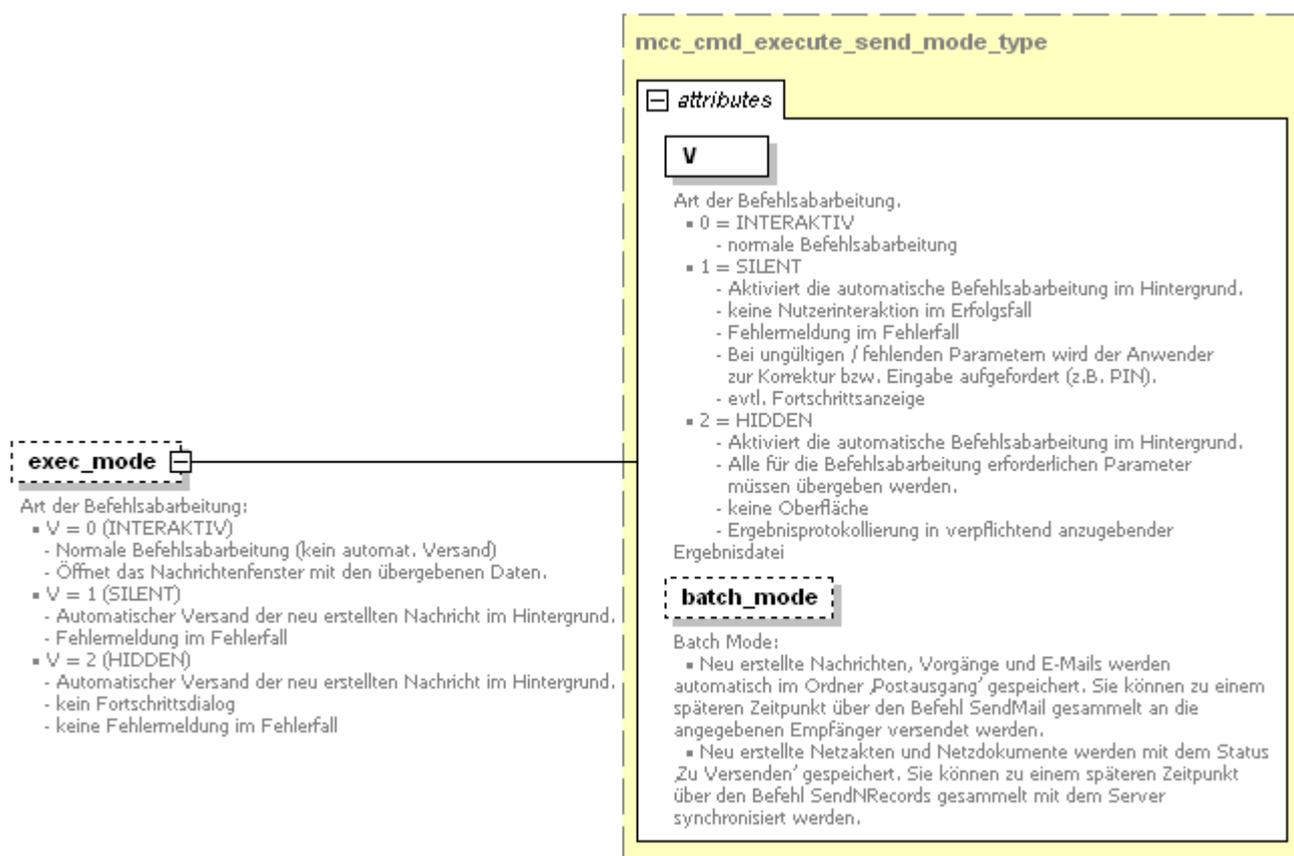


Abbildung 14 CREATE_MAIL.exec_mode

Das Element besteht aus den Attributen @V und @batch_mode. Im Attribut @V wird der Typ der Befehlsabarbeitung codiert abgelegt. Das Attribut kann dabei die folgenden Werte annehmen:

- **0 = INTERAKTIV** (default)
 Normale Befehlsabarbeitung mit Nutzerinteraktion.
 Die neu erstellte Nachricht wird mit den übergebenen Parametern in einem Nachrichtenfenster zur weiteren Bearbeitung geöffnet (bearbeiten, vervollständigen, speichern, versenden, ...).
- **1 = SILENT**
 Aktiviert den automatischen Versand der Nachricht im Hintergrund. Wurden alle erforderlichen Parameter übergeben, so wird die Nachricht ohne Nutzerinteraktion versendet. Der Versand der Nachricht wird über eine Fortschrittsanzeige visualisiert.

Bei ungültigen oder fehlenden Parametern (z.B. fehlender / unbekannter Empfänger) wird die neu erstellte Nachricht zur weiteren Bearbeitung bzw. Korrektur in einem Nachrichtenfenster geöffnet. Fehlende Nutzerdaten (z.B. fehlendes GMC PaDok Benutzerkonto, fehlende D2D-PINs) werden vom Anwender an geeigneter Stelle abgefragt.

Auftretende Fehler werden über eine entsprechende Fehlermeldung angezeigt.

- **2 = HIDDEN**

Aktiviert den automatischen Versand der Nachricht im Hintergrund ohne jegliche Programmoberfläche. Alle für die Erstellung und den Versand der Nachricht erforderlichen Parameter müssen der Kommandodatei übergeben werden.

Bei ungültigen / fehlenden Parametern und bei auftretenden Fehlern erfolgt keine Fehlermeldung. Die Ergebnis- bzw. Fehlerprotokollierung erfolgt über eine verpflichtend anzugebene Ergebnisdatei (siehe Kapitel 3.2.3).

Über das optionale Attribut *@batch_mode* kann festgelegt werden, dass die D2D-Nachricht nicht sofort an den/die Empfänger versendet werden soll. Sie wird stattdessen im Postausgang des angemeldeten Benutzerkontos gespeichert und kann zu einem späteren Zeitpunkt über den Befehl *SEND_MAIL* gesammelt versendet werden.

Als Beispiel für den automatischen Versand der D2D-Nachricht mit Abfrage fehlender Parameter sei hier der folgende Code angegeben.

```
<exec_mode V="1" batch_mode="false" />
```

XML-Code 10 CREATE_MAIL.exec_mode

3.4.5.1 Kommandoabarbeitung im „HIDDEN“-Modus

Wie bereits erläutert müssen alle, für den auszuführenden Befehl im HIDDEN-Modus benötigten Parameter der Kommandodatei übergeben werden.

Dazu gehören die Daten zur Programmanmeldung und spezielle befehlsabhängige Parameter.

Daten zur Programmanmeldung (siehe Kapitel 3.2.1):

- Name des GMC PaDok Anwendungsprofiles (falls Profile eingerichtet wurden)
- Benutzerkonto (Teilnehmer-ID und Benutzerkennwort falls vergeben)
- D2D-PIN's (für D2D-bezogene Befehle)

Im Fall des Befehls *CREATE_MAIL* sind folgende befehlspezifische Parameter zwingend anzugeben:

- Empfänger der Nachricht (siehe Kap. 3.4.1)
- Hauptdokument der Nachricht (Dateipfad und Kommentar – siehe Kap. 3.4.3)

4 Anhang

4.1 Tabellenverzeichnis

| | | |
|-----------|---|----|
| Tabelle 1 | Das type-Attribut im Element <i>import_file</i> | 14 |
| Tabelle 2 | Dokumententypen..... | 18 |
| Tabelle 3 | Dateiformate | 18 |
| Tabelle 4 | Signaturformate | 18 |

4.2 Abbildungsverzeichnis

| | | |
|--------------|--|----|
| Abbildung 1 | Grundstruktur der Kommandodatei (mcc_cmd_file)..... | 6 |
| Abbildung 2 | mcc_cmd_file.user | 7 |
| Abbildung 3 | mcc_cmd_file.sync_event..... | 8 |
| Abbildung 4 | mcc_cmd_file.result_file | 9 |
| Abbildung 5 | Allgemeiner Kommandoaufbau (mcc_cmd_base_type)..... | 10 |
| Abbildung 6 | Bisher unterstützte GMC PaDok Befehle (command_list) | 11 |
| Abbildung 7 | Das Kommando CREATE_MAIL..... | 12 |
| Abbildung 8 | CREATE_MAIL.receiver | 13 |
| Abbildung 9 | CREATE_MAIL.patient | 13 |
| Abbildung 10 | CREATE_MAIL.patient.patient_id | 14 |
| Abbildung 11 | CREATE_MAIL.patient.import_file | 15 |
| Abbildung 12 | CREATE_MAIL.maindocument | 16 |
| Abbildung 13 | CREATE_MAIL.attachment..... | 19 |
| Abbildung 14 | CREATE_MAIL.exec_mode | 20 |

4.3 XML-Code-Verzeichnis

| | | |
|-------------|---------------------------------------|----|
| XML-Code 1 | Aufbau der Kommandodatei | 5 |
| XML-Code 2 | mcc_cmd_file.user | 7 |
| XML-Code 3 | mcc_cmd_file.sync_event..... | 8 |
| XML-Code 4 | mcc_cmd_file.result_file | 9 |
| XML-Code 5 | Das Kommando CREATE_MAIL..... | 12 |
| XML-Code 6 | CREATE_MAIL.receiver | 13 |
| XML-Code 7 | CREATE_MAIL.patient.patient_id | 14 |
| XML-Code 8 | CREATE_MAIL.patient.import_file | 15 |
| XML-Code 9 | CREATE_MAIL.main_document | 16 |
| XML-Code 10 | CREATE_MAIL.exec_mode | 21 |