

# PeriData LINKMODUS: die Schnittstelle zur Praxis-Software

## Vorwort

PeriData kann in einem speziellen Modus, dem LINKMODUS, aufgerufen werden, und arbeitet dann als Server für einen Client. Der Client ist in der Regel ein Praxis-Software-Programm. PeriData arbeitet im LINKMODUS patientenbezogen (separate Datei für jeden Patient ) und benötigt dafür Informationen vom Client. PeriData liefert auch Informationen an den Client über die Untersuchungen, damit diese in der Patientenakte eingetragen werden. Ausserdem benötigt PeriData Informationen über den Modus der Zusammenarbeit (es gibt 2 verschiedene LINKMODI). Der Austausch der Informationen / Daten erfolgt über die Kommandozeile und über Dateien.

## Inhalt

1. Parameter-Linkmodus (ONEPAT-Linkmodus - Version 1.3.1)
2. GDT-Linkmodus (Version 1.1.0)
3. Ergänzende Steuerbefehle (Version 1.4.0)

## 1. Parameter-Linkmodus

### Version 1.3.1

implementiert in PeriData for Windows ab Version 1.3.1 - 09.11.1999

Autoren: J. Weber, G. Yannikis

Copyright: PeriData Software GmbH, Am Simonishof 11, 50354 Hürth

### 1.1. Allgemeines

Im Linkmodus arbeitet PeriData als Server für einen Client. Der Client ist in der Regel ein Praxis-Software-Programm. PeriData arbeitet im Linkmodus patientenbezogen (separate Datei für jeden Patient ) und benötigt dafür Informationen vom Client. PeriData liefert auch Informationen an den Client über die Untersuchungen, damit diese in der Patientenakte eingetragen werden. Ausserdem benötigt PeriData Informationen über den Modus der Zusammenarbeit (es gibt 2 verschiedene Linkmodi).

Im Parameter-Linkmodus erfolgt der Austausch der Informationen / Daten über die Kommandozeile (Empfänger: PeriData) und über Dateien (Empfänger: Praxissoftware) .

### 1.2. Kommandozeile

Die Reihenfolge der Parameter auf der Kommandozeile spielt keine Rolle. Die maximale Anzahl der Zeichen darf 255 nicht übersteigen (inklusive der Leerzeichen zwischen den Parametern). Gross/Kleinschreibung spielt mit Ausnahme des Parameters /ansi:c bzw. /ANSI:C keine Rolle.

Kommandozeilen-Parameter	Bedeutung
/MODE:<Modus>	Modus = ONEPAT
/CLIENT:<Clientname>	Clientname = Name des Praxisprogrammes (z.B. SUPERMEDISOFT)
/NAME:< Name >	Name = Familienname des Patienten
/FIRSTNAME:< Vorname >	Vorname = Titel + Vorname(n) des Patienten
/BORN:< Geburtsdatum >	Geburtsdatum = ISO-Format: JJJJMMTT ohne Punkte

/FILE:< Filename >	Filename = Filename mit vollständigem Pfad der PeriData-Datei (PHF-Datei) des Patienten
/READONLY	Optional, sperrt die Datenbank fuer Schreibzugriffe
/ANSI:F	Text in uebergebenen Files im ANSI-Code (Umlaute)
/ANSI:C	Kommandozeile im ANSI-Code (Umlaute)

### Parameter /MODE

Der Modus ONEPAT (ein Patient) bedeutet, daß das spezifizierte File nur für 1 Patienten vorgesehen ist. Ferner sollen ankommende Daten nur in dieses Patienten-File gespeichert werden. Die Namen in diesem Patienten-File werden überschrieben. Andere Dateien können geöffnet werden, um beispielsweise Daten zu transferieren (alte Daten aus anderen Dateien importieren über das Clipboard...). Wenn das Patienten-File geschlossen wurde und Daten über Kabel ankommen, wird das File wieder geöffnet.

### Parameter /CLIENT

Der Name des Clients, also die Praxissoftware oder Kliniksoftware. Dieser Name erscheint im PeriData-Programmfenster in eckigen Klammern.

### Parameter /NAME

Familiennamen des Patienten. Leerzeichen sind nicht erlaubt. Unterstriche werden nach dem Auslesen der Kommandozeile in Leerzeichen umgewandelt. Titel gehören zum Vornamen, sonstige Namenszusätze zum Namen, da sie manchmal zusammengeschrieben werden.

*Beispiel (Dr. med. Hans von Humboldt): /NAME:von\_Humboldt*

### Parameter /FIRSTNAME

Titel + Vorname(n) des Patienten. Unterstriche werden nach dem Auslesen der Kommandozeile in Leerzeichen umgewandelt. Titel gehören zum Vornamen.

*Beispiel (Dr. med. Hans von Humboldt): /FIRSTNAME:Dr.\_med.\_Hans*

### Parameter /BORN

Geburtsdatum des Patienten im ISO-Format ohne Punkte: JJJJMMTT

*Beispiel: 19670430*

### Parameter /FILE

String mit dem Namen des Datenfiles (PeriData Datenformat). Jeder Patient besitzt ein eigenes Datenfile. Der Pfad muss vollständig, beginnend mit \, angegeben sein. Der Pfad muß vorhanden sein, die Datei muß noch nicht vorhanden sein (wenn zu dem Pat. noch keine Gesichtsfeld-Daten abgespeichert sind). Der Dateiname kann ein patientenspezifischer Name sein, daß die einzigartige Patientenummer enthält.

*Beispiel A: /FILE:C:\DATEN\FREMD\PERIDATA\004312.PHF*

Oder es kann eine temporäre Datei sein, die der Client erzeugt und in die er die Daten aus einer anderen Quelle lädt, z.B. aus seiner eigenen Datenbank:

*Beispiel B: /FILE:C:\TEMPORAER\TEMP.PHF*

### Parameter /READONLY

Dieser Parameter ist optional. Wenn der Parameter /READONLY angegeben wird, dann wird die im Parameter /File spezifizierte Datei nur zum Lesen geöffnet. In dieser Datei können dann keine Änderungen der Untersuchungen mehr vorgenommen werden.

Beachten Sie bitte: Die im Parameter /File spezifizierte Datei wird unter Umständen durch PeriData auch gesperrt ohne die Angabe dieses Parameters. Dieser Fall tritt dann ein, wenn die Datei im Netzwerk bereits von einem anderen Prozess geöffnet wurde !

**Parameter /ANSI:F**

Dieser Parameter ist optional. Es werden alle Zeichen in den uebergebenen Files (GDT-Dateien) aus dem erweiterten ANSI-Zeichensatz in den OEM-Zeichensatz konvertiert. Diese Konvertierung wirkt sich auf die Umlaute im Patientennamen aus.

**Parameter /ANSI:C (auch erlaubt: /ansi:c)**

Dieser Parameter ist optional. Es werden alle Zeichen auf der Kommandozeile aus dem erweiterten ANSI-Zeichensatz in den OEM-Zeichensatz konvertiert. Die Konvertierung wird VOR der Auswertung der Kommandozeile gemacht. Diese Konvertierung wirkt sich die Umlaute im Patientennamen, aber auch auf Dateinamen mit Umlauten aus.

**Bemerkung zu beiden /ANSI - Parametern**

Anstelle dieser beiden Parameter kann auch die Option 'Umlaute wandeln' im Programm eingeschaltet werden (Menu 'Extras' - Befehl 'Optionen' - Karteikarte 'Allgemein' ab Version 1.5). Diese Option wirkt so wie beide Parameter zusammen. In Spezialfaellen, wenn der Programmaufruf und die Benachrichtigungsdatei in unterschiedliche Zeichensatze verwenden, kann die Anpassung nur ueber die /ANSI-Parameter erfolgen.

**1.3. Verhalten von PeriData****Verhalten beim Mehrfach-Start von PeriData auf einem PC**

Peridata kann vom Client mehrfach gestartet werden. Von PeriData existiert jedoch nur immer eine Instanz im Speicher. Wenn im Praxis-Programm der Patient wechselt, muß es PeriData erneut mit neuen Kommandozeilen-Parametern aufrufen. Die neuen Parameter werden wieder ausgewertet. PeriData schliesst dann automatisch alle seine offenen Dateien und öffnet die Gesichtsfeld-Datei des neuen Patienten.

**Verhalten bei fehlendem Daten-File (.PHF-File)**

Wenn die im Parameter /File spezifizierte Datei fehlt, wird sie automatisch neu erzeugt. Die neue Datei ist leer. Die zugehörige PHD-Datei wird ebenfalls erzeugt.

**Verhalten bei fehlendem Sortier-File (.PHD-File)**

Wenn das PHD-File fehlt, wird es aus dem PHF-File erzeugt.

**Verhalten beim PeriData-Start mit Kopien von PHF-Files aus dem Client-System**

Üblich ist die Nennung des Original-Pfads und Original-Files als File-Parameter. Lokale Kopien aus dem Client-System sind problematisch, da sich beim Mehrfach-Zugriff auf den gleichen Patienten Aktualitäts-Widersprüche ergeben können. Die Funktionalität von PeriData beim Mehrfach-Zugriff (s.u.) ist nicht mehr vorhanden. Von dieser Konstellation ist daher dringend abzuraten, da sie vom Client-System zusätzlichen Programmier- und Testaufwand verlangt.

**Verhalten beim PeriData beim Mehrfach-Zugriff auf die gleiche Daten-Bank (.PHF-File und .PHD-File)**

Bei üblicher Konstellation (Client-System nennt für jeden Patienten an allen Arbeitsplätzen eine einzigartige Original-File-Kennung als File-Parameter) kann es zum gleichzeitigen Zugriff auf die gleiche Datenbank (PHF-File und PHD-File) kommen. Dazu gibt es zwei Möglichkeiten: 1. Standard: Der erste User erhält Lese- und Schreibrechte. Jeder weitere User erhält nur Schreibrechte. Letzteres wird gemeldet und ist durch ein Kreuz über dem PeriData-Symbol im Untersuchungslisten-Fenster gekennzeichnet. 2. "Immer Read-Only": Wenn ein User in die Option "nur Lesefunktion" gewählt hat, öffnet er die Dateien im Read-Only-Modus. Das ermöglicht dem nächsten User, die Schreibrechte zu erhalten. Sinnvollerweise sollten alle Arbeitsplätze außer denen, die Daten vom Perimeter empfangen, in diesen Modus versetzt werden.

**1.4 Fehlermeldungen von PeriData beim Aufruf**

Fehlermeldungen bedeuten Fehler bei der Aufrufprozedur, entweder bei den Parametern oder bei den Dateien

**Fehler 1**

Parameter /MODE fehlt oder unlesbar. Nur ONEPAT eingeben

**Fehler 2**

Parameter /CLIENT fehlt oder unlesbar. Ein Client muss immer angegeben werden.

**Fehler 3**

Parameter /FIRSTNAME fehlt oder unlesbar. Der Vorname muss immer angegeben werden.

**Fehler 4**

Parameter /NAME fehlt oder unlesbar. Der Name muss immer angegeben werden.

**Fehler 5**

Parameter /BORN fehlt oder unlesbar. Das Datum muss mit 8 Zeichen angegeben werden: JJJJMMTT. Ein Jahr unter 1800 ist sicher ein Fehler.

**Fehler 6**

Parameter Vorname, Name oder Geburtsdatum nicht sinnvoll

**Fehler 7**

Parameter /FILE fehlt oder nicht sinnvoll. Vollständiger Pfad mit allen Backslashes notwendig, also C:\Perimetrie\Daten\2456.phf

**Fehler 10**

Perimetrie-Datei konnte nicht geöffnet werden. Datei fehlt oder keine Zugriffsrechte.

**Fehler 11**

Perimetrie-Datei konnte nicht erzeugt werden. Verzeichnis fehlt oder keine Zugriffsrechte.

**Fehler 14**

Verzeichnis für die Perimetrie-Datei fehlt oder keine Zugriffsrechte.

**Fehler 15**

Message-Datei konnte nicht generiert werden.

### ***1.5. Inhalt der MES-Datei beim Melden von Untersuchungen an den Client (PeriData schreibt)***

Jedesmal, wenn Untersuchungen von einem Perimeter empfangen werden oder auch ältere Untersuchungen aus einer anderen Datei in die aktive Patientendatei kopiert werden, meldet PeriData jede neue Untersuchung in Form einer Datei-gebundenen Nachricht (Verzeichnis muss vom Client gepollt werden!). Für jede neue Untersuchung wird eine MES-Datei erzeugt. Sie liegt im gleichen Verzeichnis, in dem die übergebene Datei (Parameter /FILE:) liegt.

Der Client ist verantwortlich für das Löschen dieser Dateien, nachdem er sie gelesen und ausgewertet hat. Files mit mehr als 1 Tag Datumsdifferenz werden von PeriData in Abständen gelöscht. Wenn also ein Programm diese Funktion nicht unterstützt, kümmert sich PeriData um die „Reinigung“ des Verzeichnisses

Die Namen der erzeugten Dateien sind fortlaufende Nummern (z.B. 00000000.mes, 00000001.mes usw.) im 8.3-Format. Die Zaehlung beginnt beim echten Neustart wieder von vorne, wobei nicht gelöschte Dateien geschont und uebersprungen werden. Die erzeugten MES-Dateien enthalten nur die Kenndaten einer Untersuchung, nicht die eigentlichen Daten. Jede der folgenden Informationen sind in einer Zeile mit Line feed und Carriage return.

<b>Zeileninhalt</b>	<b>Erläuterung</b>
/PROGRAM: PERIDATA	Information über das erzeugende Programm
/NAME:< Name >	Name = Familienname des Patienten

/FIRSTNAME:< Vorname >	Vorname = Titel + Vorname(n) des Patienten
/BORN:< Geburtsdatum >	Geburtsdatum (ISO-Format: JJJMMTT ohne Punkte)
/FILE:< Filename >	Filename = Filename OHNE Pfad der Untersuchungsdaten (PHF-Datei) - enthält evtl. die Patienten-Nummer
/DATE:	Untersuchungs-Datum (ISO-Format: JJJMMTT ohne Punkte)
/TIME:	Untersuchungs-Zeit (Format: HHMM)
/EYE:	Untersuchtes Auge (Lateinisches Format: OD / OS)
/EXAM:	Untersuchungsinfo (Format: <Auge>: <Geraet> <Untersuchungsprogramm> [30-50 Zeichen]) (Beispiel RA: HFA2 30-2 SCHWELLENTTEST)
/INDICES:	Untersuchungsergebnis Indizes (Beispiel: MD: -10.2 DB / FD: -77 % / PSD: 5.2 DB)
/RELIABILITY:	Untersuchungsergebnis Zuverlässigkeit (Beispiel: FIXLOSS 0% / FPOS 6% / FNEG 0%)

Beispiel für die Datei:

```

/PROGRAM: PERIDATA
/NAME: von_Humboldt
/FIRSTNAME: Dr._med._Hans
/BORN: 19670430
/FILE: C:\TEMPORAER\TEMP.PHF
/DATE: 19991127
/TIME: 1142
/EYE: OD
/EXAM: HFA2 30-2 SCHWELLENTTEST
/INDICES: MD: -10.2 DB / FD: -77 % / PSD: 5.2 DB
/RELIABILITY: FIXLOSS 0% / FPOS 6% / FNEG 0%
```

### 1.6. Auswertung der MES-Datei im Client

Die MES-Datei enthält nur die Kenndaten (Untersuchungs-Info, Datum, Uhrzeit) einer Untersuchung. Sinn ist die Eintragung dieser Kenndaten in eine Zeile der Patientenakte. Die Auswertung der übermittelten Daten und die Gestaltung der Zeile ist Sache des Client. Denkbar:

- /PROGRAM: PERIDATA wird für eine Definition der Zeilenart verwendet (z.B. „Peri“)
- /DATE wird für das Datum der Zeile / die Einordnung in der Patientenakte verwendet
- /TIME wird am Anfang (oder am Ende) der Zeile in Klammern angezeigt
- /EXAM wird Zeilen-Inhalt

Beispiel der Patientenakte für Untersuchungen beider Augen (2 MES-Dateien):

--	--	--

---

27.11.1999	Peri	(11:42) OD: HFA2 30-2 SCHWELLENTTEST
27.11.1999	Peri	(12:05) OS: HFA2 30-2 SCHWELLENTTEST

## 2. GDT-Linkmodus

### Version 1.1.0

implementiert in PeriData for Windows ab Version 1.1.0 - 30.04.1999

Autoren: J. Weber, G. Yannikis

Copyright: PeriData Software GmbH, Am Simonishof 11, 50354 Hürth

### 2.1. Referenz

Die GDT-Schnittstelle von PeriData beruht auf dem Dokument:

Anbindung von medizinischen Meßgeräten (GDT - Gerätedaten-Träger)  
Schnittstellenbeschreibung zum  
systemunabhängigen Datentransfer zwischen  
Praxis-EDV-Systemen und Meßgeräten, Version 1.0 (12/95)

herausgegeben vom

Qualitätsring Medizinische Software (QMS)  
c/o Kassenärztliche Vereinigung Hessen  
Bezirksstelle Darmstadt  
Wilhelminenplatz 7  
64283 Darmstadt

### 2.2. Allgemeines

Im Linkmodus arbeitet PeriData als Server für einen Client. Der Client ist in der Regel ein Praxis-Software-Programm. PeriData arbeitet im Linkmodus patientenbezogen (separate Datei für jeden Patient ) und benötigt dafür Informationen vom Client. PeriData liefert auch Informationen an den Client über die Untersuchungen, damit diese in der Patientenakte eingetragen werden. Ausserdem benötigt PeriData Informationen über den Modus der Zusammenarbeit (es gibt 2 verschiedene Linkmodi).

Der Austausch der Informationen / Daten erfolgt über die Kommandozeile, Messages und über Dateien im GDT-Format. Die Definition der Kommandozeile und der Messages ist PeriData-spezifisch. Das Format der GDT-Dateien folgt der GDT-Definition 1.0.

### 2.3. Kommandozeile

Die Reihenfolge der Parameter auf der Kommandozeile spielt keine Rolle. Die maximale Anzahl der Zeichen darf 255 nicht übersteigen (inklusive der Leerzeichen zwischen den Parametern).

Gross/Kleinschreibung spielt mit Ausnahme des Parameters /ansi:c bzw. /ANSI:C keine Rolle.

Kommandozeilen-Parameter	Bedeutung
/MODE:<Modus>	Modus = GDT: PeriData startet im GDT-LINKMODUS (patientenbezogen, Client-Informationen werden aus GDT-Files gelesen) Modus = GDT_MSG: wie vor, jedoch mit dem Austausch von Windows-Messages
/CLIENT:<Clientname>	Clientname = ID des Praxisprogrammes (z.B. SUPERMEDISOFT)
/GDT:< Pfadname >	Pfadname = Pfad für die GDT-Dateien (Import und Export)
/DATA:< Pfadname >	Pfadname = Pfad der PeriData-Gesichtsfeld-Dateien

/ANSI:F	Text in GDT-Files im ANSI-Code (Umlaute)
/ANSI:C	Kommandozeile im ANSI-Code (Umlaute)

### Parameter /MODE

Wenn der Modus GDT heißt, liest PeriData sofort die PERI???.??? im GDT-Pfad (s.u.). Wenn der Modus GDT\_MSG heißt, wartet PeriData auf den Befehl GDT\_OpenPatient. Später werden nur im Modus GDT\_MSG auch die Windows-Messages GDT\_NewData usw. geschickt.

### Parameter /CLIENT

Der Name des Clients, also die Praxissoftware oder Kliniksoftware. Dieser Parameter hat 2 Bedeutungen:

1. Der Name erscheint im PeriData-Programmfenster in eckigen Klammern,
2. Die ersten 4 Buchstaben des Client-Namens bestimmen den Namen der GDT-Dateien (siehe unten).

### Parameter /GDT

String mit dem Pfad der GDT-Dateien

*Beispiel:* C:\INTERFACE\GDT\PERIDATA

PeriData liest aus dem Pfad die Datei PERI???.??? den Patienten-Stammdaten und schreibt seine Untersuchungsinfo-Files in das gleiche Verzeichnis (Zum Namen der Files siehe im entsprechenden Kapitel)

**Achtung: Der Pfad muss vollstaendig, mit \ hinter dem Laufwerk, angegeben sein.**

**Achtung: Der Pfadname muß identisch mit dem Eintrag in der Geräte-Installation des Clientprogramms sein (Exportdatei UND Importdatei).**

**Achtung: Aus Gründen der Sicherheit ist es empfehlenswert, das jeder Client (Arbeitsplatz) ein eigenes Verzeichnis verwendet. Bei einem gemeinsamen Verzeichnis besteht die Gefahr, daß mehrere für PeriData bestimmte GDT-Dateien gleichzeitig in dem Verzeichnis liegen und eine PeriData-Instanzen die für die andere Instanz bestimmte GDT-Datei öffnet (mit den falschen Patientendaten). Alternativ kann der Client ein festes Datei-Extend vereinbaren, z.B. bei dem Client "SUPERMEDISOFT" den Namen PERISUPE.GDT. Solange die erste PeriData-Instanz die Datei noch nicht gelesen und danach gelöscht hat, kann der zweite Client die Datei nicht schreiben und führt auch keinen PeriData-Call aus.**

### Parameter /DATA

String mit dem Pfad für die Untersuchungsdaten (PeriData-Format: \*.phf und \*.phd).

*Beispiel:* C:\DATEN\FREMD\PERIDATA

PeriData legt seine Daten in diesem Verzeichnis ab.

**Achtung: Der Pfad muss vollstaendig, mit \ hinter dem Laufwerk, angegeben sein.**

**Achtung: Das Verzeichnis muß vom Client angelegt worden sein.**

### Parameter /ANSI:F

Dieser Parameter ist optional. Es werden alle Zeichen in den uebergebenen Files (GDT-Dateien) aus dem erweiterten ANSI-Zeichensatz in den OEM-Zeichensatz konvertiert. Diese Konvertierung wirkt sich auf die Umlaute im Patientennamen aus.

### Parameter /ANSI:C (auch erlaubt: /ansi:c)

Dieser Parameter ist optional. Es werden alle Zeichen auf der Kommandozeile aus dem erweiterten ANSI-Zeichensatz in den OEM-Zeichensatz konvertiert. Die Konvertierung wird VOR der Auswertung der Kommandozeile gemacht. Diese Konvertierung wirkt sich die Umlaute im Patientennamen, aber auch auf Dateinamen mit Umlauten aus.

### Bemerkung zu beiden /ANSI - Parametern

Anstelle dieser beiden Parameter kann auch die Option 'Umlaute wandeln' im Programm eingeschaltet werden (Menu 'Extras' - Befehl 'Optionen' - Karteikarte 'Allgemein' ab Version 1.5).



Diese Option wirkt so wie beide Parameter zusammen. In Spezialfaellen, wenn der Programmaufruf und die Benachrichtigungsdatei in unterschiedliche Zeichensaetze verwenden, kann die Anpassung nur ueber die /ANSI-Parameter erfolgen.

**Definition der Pfade**

Wenn PeriData von einem Client gestartet wird, dann bestimmt der Client, wo seine GDT-Dateien liegen, und wo die PHF-Dateien liegen sollen, damit sie von seiner Datensicherung erfaßt werden. Die Patientenummer in der PHF-Datei wird erst durch die Verbindung mit dem Client (Praxis-Software) eindeutig (in einem anderen Praxis-Software-Programm bedeutet die Nummer einen anderen Patienten). Daraus folgt:

Diese beiden Pfade gehören dem Client und werden von ihm definiert und gespeichert, am besten beim Eintragen von PeriData in seine Geraete-Liste.

**Verhalten beim Mehrfach-Start von PeriData**

Peridata kann vom Client mehrfach gestartet werden. Von PeriData existiert jedoch nur immer eine Instanz im Speicher. Diese Eigenschaft wird genutzt, wenn die Schnittstelle ohne Windows-Messages betrieben wird (Modus GDT). Wenn im Praxis-Programm der Patient wechselt, wird PeriData erneut mit neuen Kommandozeilen-Parametern aufgerufen werden. Die neuen Parameter werden wieder ausgewertet. In der Regel sind nicht die Parameter neu, sondern nur die GDT-Datei und die darin enthaltene Patienten-Nummer. PeriData wertet nach diesem Aufruf die neue GDT-Datei aus. PeriData schliesst dann automatisch alle seine offenen Dateien und öffnet die Gesichtsfeld-Datei des neuen Patienten.

Bemerkung: Dieses Verhalten ist nur für den Aufruf von PeriData mit CREATE PROCESS getestet.

**2.4. Windows-Messages**

Die Windows-Messages dienen der verzögerungsfreien Kommunikation zwischen Client und Server. Im Gegensatz zur Kommunikation über Programmstart und Kommandozeilen erlauben Windows-Messages, daß das Server-Programm PeriData offen bleibt, wenn ein Patient wechselt (bei PeriData mit einem Trick verhindert). **Windows-Messages werden von PeriData nur ausgewertet, wenn es mit dem Parameter /MODE : GDT\_MSG gestartet wurde.**

Message	Client	PeriData
GDT_OpenPatient	SENDET MESSAGE wenn der Client den Patienten wechselt (evtl. nur, wenn PeriData vom Benutzer aktiviert wird, dann aber ClosePatient verwenden!) nachdem der die Datei PERIDATA.GDT geschrieben hat!	EMPFÄNGT MESSAGE PeriData schließt alle Dateien. PeriData liest die PERIDATA.GDT. PeriData nimmt die Patienten-Stammdaten als „aktiven Patient“ PeriData öffnet die Datei <Patientennummer>.PHF. Wenn sie fehlt, wird sie neu erzeugt.
GDT_ClosePatient	SENDET MESSAGE wenn der Client den Patienten schließt / wechselt.	EMPFÄNGT MESSAGE PeriData schließt die Datei des aktiven Patienten.
GDT_PatientWasClosed	EMPFÄNGT MESSAGE Client kann nun über erneutes OpenPatient / SendAllData / ClosePatient sich die Daten dieses „beendeten“ Patienten holen	SENDET MESSAGE wenn das Datei-Fenster des aktiven Patienten vom Benutzer geschlossen wurde. (Keine Meldung nach Schliessen mittels ClosePatient)
GDT_ExitProgram	SENDET MESSAGE wenn er sich selbst schließt	EMPFÄNGT MESSAGE PeriData schließt alle Dateien. PeriData beendet sich selbst.
GDT_NewData	EMPFÄNGT MESSAGE	SENDET MESSAGE

	Client liest <Nummer>.GDT Client nimmt die Untersuchungsinfo und schreibt sie in seine Patienten-Akte. Client löscht <Nummer>.GDT	wenn neue Untersuchungsdaten in der Datei des aktiven Patienten gespeichert wurden nachdem er die Datei <Nummer>.GDT mit dem Untersuchungsinfo geschrieben hat
GDT_SendAllData	SENDET MESSAGE wenn er alle Untersuchungsinfos des aktiven Patienten haben will (z.B. vor ClosePatient)	EMPFÄNGT MESSAGE PeriData erzeugt für jede Untersuchung in der Datei des aktiven Patienten eine GDT-Datei mit dem Untersuchungsinfo sendet für jede geschriebene GDT-Datei ein GDT_NewData

**weitere Spezifikationen**

Message-Typ: SendMessage

Parameter 1 = hWnd ist das Windows-Handle des Message-Fensters des Empfängers der Message. Falls der Client schickt, muß er das PeriData-Message-Fenster suchen. Es wird identifiziert als TopLevel mit der Caption „PeriData Shell“. Falls PeriData schickt, muß es das Client-Fenster verwenden. Dieses wird bei der ersten Message (GDT\_OpenPatient) vom Client als Parameter 4 mitgegeben (s.u.). Diese Regelung ist notwendig, da Peridata von verschiedenen Clients bzw. Prozessen mit verschiedenen Message-Fenstern aufgerufen werden könnte. Es läßt außerdem dem Client die freie Wahl für die Bezeichnung seines Message-Fensters.

Parameter 2 = wParam enthält die Windows-ID, die die Prozedur RegisterMessage zurückgibt

Parameter 3 = lParam (in Windows 95/NT Doppelwort) enthält die Patient-ID beim Client. Notwendig bei den Messages GDT\_ClosePatient, GDT\_PatientWasClosed und GDT\_SendAllData, die kein „begleitendes“ GDT-File haben, um den Patienten eindeutig zu identifizieren.

Parameter 4 = lParam ist das Windows-Handle des Message-Fensters des Senders der Message (wegen Rückmessage).

**2.5. Inhalt der GDT-Datei beim Aufruf von PeriData (Client schreibt)**

Bei Aufruf mit /MODE:GDT oder nach Aufruf mit /MODE:GDT\_MSG bei einer entsprechenden Windows-Message liest PeriData im GDT-Verzeichnis eine Datei mit den Stammdaten, die das Praxisprogramm (der Client) geschrieben hat und wertet sie aus.

**Name der Datei:** Die vom Client exportierte Datei muß nach GDT mit den 4 Buchstaben der Empfänger-ID (hier immer PERIDATA) beginnen, dann kommen 4 Buchstaben der Sender-ID (aus dem Parameter /CLIENT). Extend kann GDT oder eine Zahl (001, 002..) sein. Bei einem Client namens SUPERMEDISOFT heißt die Datei PERISUPE.GDT oder PERISUPE.001. PeriData ignoriert das Extend.

**Inhalt der Datei:** Die erste Zeile bestimmt die Satzart. Im Aufruf von PeriData muss hier der Wert 6301 ode 6302 oder 6311 (6301=Stammdaten übermitteln; 6302=Daten einer Untersuchung anfordern; 6311= Daten einer Untersuchung zeigen) verwendet werden. PeriData reagiert auf allerdings gleichartig: Das Datenbankfenster öffnet sich, falls Daten vorhanden sind, werden alle Daten im Serie-Fenster gezeigt.

Die Reihenfolge der restlichen Parameter spielt keine Rolle. PeriData verlangt mindestens die folgenden Parameter in der GDT-Datei:

GDT-Zeile (Länge + Feldkennung + Wert)	Feldinhalt (Wert)
xxx 8000 <Wert>	Satzart (immer 6301 oder 6302 oder 6311))
xxx 8100 <Wert>	Dateilänge (total)
xxx 8315 <Wert> *ignoriert*	Empfänger-ID (PERIDATA)

xxx 8316 <Wert> *ignoriert*	Sender-ID (<Clientname>, wie beim Aufruf als Kommandozeilenparameter übergeben, 1-8 Zeichen z.B. SUPERMED)
xxx 9218 <Wert> *ignoriert*	GDT-Version (01.00 oder 02.00)
xxx 3000 <Wert>	Patienten-Nummer beim Client (Format: 1-8 Stellen)
xxx 3101 <Wert>	Nachname des Patienten
xxx 3102 <Wert>	Vorname des Patienten
xxx 3103 <Wert>	Geburtsdatum des Patienten (Format: DDMMJJJJ)
alle weiteren Felder (auch Satzlänge, Version) werden nicht ausgewertet	
(Zeilenlänge wird nicht ausgewertet)	

Die Ignorierung der Felder 8315, 8316 und 9218 trägt der Realität der GDT-Schnittstellen Rechnung. Bei verbesserter Situation sollen auch diese Felder ausgewertet werden.

Die Namen der Gesichtsfeld-Dateien, die PeriData generiert, sind immer im 8.3-Format. Der Name enthält die Patientenummer, denen führende Nullen vorangestellt werden. Das Extent ist PHF. Dazu wird ein Sortier-File mit gleichem Namen und dem Extent PHD erzeugt. Beispiel: Patient 582 > 00000582.phf 00000582.phd

Bei Praxisgemeinschaften kann es zu doppelten Patientennummern kommen. Die Lösung obliegt dem Client. Die PeriData Software GmbH empfiehlt, die übermittelte Nummer nach folgendem Schema zusammenzusetzen

Ziffer 1-2: Nr. der Praxis

Ziffer 3-8: Nr. des Patienten, mit führenden Nullen ergänzt

Der Umfang beträgt dann: bis zu 99 Praxen pro Praxisgemeinschaft und bis zu 999 999 Patientennummern (also bis 1 Million). Der Client muß sich für ein Schema entscheiden und darf es nicht ändern, da sonst die Zuordnung verloren geht.

Wenn ein Parameter aus der obigen Tabelle fehlt, zeigt Peridata eine Fehlermeldung und beendet sich.

## 2.6. Inhalt der GDT-Datei beim Melden von Untersuchungen an den Client (PeriData schreibt)

Jedesmal, wenn Untersuchungen von einem Perimeter empfangen werden oder auch ältere Untersuchungen aus einer anderen Datei in die aktive Patientendatei kopiert werden, meldet PeriData jede neue Untersuchung in Form einer Datei-gebundenen Nachricht (keine ECHTE message, Verzeichnis muss gepollt werden!). Für jede neue Untersuchung wird eine GDT-Datei erzeugt. Sie liegt im gleichen Verzeichnis, aus dem PeriData GDT-Dateien liest. Der Client ist verantwortlich für das Löschen dieser Dateien (nachdem er sie gelesen und ausgewertet hat). PeriData löscht automatisch liegendebliebene Dateien, die älter als ein Tag sind (falls die Auswertungsroutine des Client fehlt oder versagt).

**Name der Dateien:** Die vom Server exportierte Datei muß nach GDT mit den 4 Buchstaben der Empfänger-ID (aus dem Parameter /CLIENT) beginnen, dann kommen 4 Buchstaben der Sender-ID (hier immer PERIDATA). Extend ist eine Zahl (001, 002 usw). Bei einem Client namens SUPERMEDISOFT heißen die Dateien SUPEPERI.001, SUPEPERI.002 usw.

Inhalt der Datei: Die erste Zeile bestimmt die Satzart. In dieser GDT-Datei muß dabei der Eintrag 8000 und 6310 in der ersten Zeile erscheinen.

**Satzart 6300 = Stammdaten anfordern wird nicht unterstützt.**

Die erzeugten GDT-Dateien enthalten nur die Kenndaten einer Untersuchung, nicht die eigentlichen Daten:

GDT-Zeile (Länge +	Feldinhalt (Wert)
--------------------	-------------------

<b>Kennung + Wert)</b>	
xxx 8000 <Wert>	Satzart ( <i>immer: 6310 = Daten einer Untersuchung übermitteln</i> )
xxx 8100 <Wert>	Anzahl Zeichen (ohne Wert, ohne CR/LF) + 2*Anzahl Zeilen = Dateilänge (total)
xxx 8315 <Wert>	Empfänger ( <i>PRAX_EDV oder &lt;Clientname&gt;, wie beim Aufruf als Kommandozeilenparameter übergeben, 1-8 Zeichen</i> )
xxx 8316 <Wert>	Sender ( <i>PERIDATA</i> )
xxx 9218 <Wert>	GDT-Version ( <i>01.00</i> )
xxx 3000 <Wert>	Patientennummer beim Client. Identisch mit der Nummer in der Stammdaten-GDT-Datei ( <i>Format: 1-8 Stellen</i> )
xxx 8402 <Wert>	Untersuchungstyp ( <i>OPTO05</i> ) <i>OPTO05 bedeutet Gesichtsfeldmessung</i>
xxx 8410 <Wert>	Testdaten-Identifikation ( <i>BEMERK</i> ) <i>BEMERK bedeutet, daß nur das Testdatum und die Uhrzeit ohne eigentliche Daten folgen</i>
xxx 8432 <Wert>	Untersuchungsdatum ( <i>Format: DDMMJJJJ</i> )
xxx 8439 <Wert>	Untersuchungszeit ( <i>Format: HHMM</i> )
xxx 6227 <Wert>	Kommentar, hier: Untersuchungsinfo ( <i>2-3 Zeilen</i> ) <i>Beispiel RECHTS: HFA2 30-2 SCHWELLENTTEST RELIABILITY: FIXLOSS 6% / FPOS 10% / FNEG 30% INDICES: MD -7.5 DB / FD: -56% / PSD: 7.2 DB</i>

## 2.7. Auswertung der Untersuchungsdaten-GDT-Datei im Client

Diese GDT-Datei enthält nur die Kenndaten (Untersuchungs-Info, Datum, Uhrzeit) einer Untersuchung. Sinn ist die Eintragung dieser Kenndaten in eine Zeile der Patientenakte. Die Auswertung der übermittelten Daten und die Gestaltung der Zeile ist Sache des Client. Denkbar:

6227 Kommentar wird Zeilen-Inhalt

8432 wird für das Datum der Zeilen / die Einordnung in der Patientenakte verwendet

8402 OPTO05 wird für eine Definition des Zeilentyps verwendet (z.B. „Peri“ oder "P" oder "G" )

8410 BEMERK wird ignoriert, ist nur eine Testdaten-Identifikation

8439 wird am Anfang der 1.Zeile in Klammern angezeigt

*Beispiel für Untersuchungen beider Augen:*

27.11.1999	Peri	(11:42) RA: HFA2 30-2 SCHWELLENTTEST
	Peri	RELIABILITY: FIXLOSS 6% / FPOS 10% / FNEG 30%
	Peri	INDICES: MD: -7.5 DB / FD: -56% / PSD: 7.2 DB
27.11.1999	Peri	(12:05) LA: HFA2 30-2 SCHWELLENTTEST
	Peri	RELIABILITY: FIXLOSS 0% / FPOS 5% / FNEG 10%
	Peri	INDICES: MD: -2.3 DB / FD: -21% / PSD: 4.6 DB

## 3. Ergänzende Steuerbefehle

### Version 1.4.0

implementiert in PeriData for Windows ab Version 1.4.0 – 15.02.2000

Autor: G. Yannikis

Copyright: PeriData Software GmbH, Auf dem Mühlenacker 40, 50354 Hürth

### 3.1 Die Kommandozeile als Befehlsschnittstelle

PeriData kann über die Kommandozeile Steuerbefehle entgegennehmen, dazu muss sich PeriData bereits im Speicher befinden. Wenn sich PeriData noch nicht im Speicher befindet, bleibt der Aufruf von PeriData ohne Wirkung, PeriData wird in diesem Fall auch nicht gestartet (-> beim ersten Start von PeriData darf sich also kein Parameter /Command: auf der Kommandozeile befinden!). Wird auf der Kommandozeile der Parameter /COMMAND: übergeben, werden alle anderen Parameter (z.B. /MODE: oder /CLIENT:) ignoriert. Alle zurzeit implementierten Befehle sind in Abschnitt 3.2 erläutert.

#### 3.1.1 Befehle

Befehle (Kommandos) werden mit dem Parameter /COMMAND: eingeleitet. Folgende Befehle existieren:

Zeileninhalt	Erläuterung
/COMMAND:CLOSE_APP	Beendet PeriData
/COMMAND:POPUP_APP	Stellt PeriData in den Vordergrund
/COMMAND:CLOSE_WND	Schliesst alle Fenster (keine Dialoge !)
/COMMAND:CLOSE_PAT	Schliesst alle Fenster und beendet die Sitzung des Patienten

Beim Befehl 'CLOSE\_WND' werden alle offenen Fenster geschlossen. Treffen erneut Daten über die Schnittstelle vom Perimeter ein, wird die Datei des zuletzt gesetzten Patienten wieder geöffnet und die Daten werden in diese Datei abgelegt. Beim Befehl 'CLOSE\_PAT' werden ebenfalls alle Fenster geschlossen, zusätzlich wird die Sitzung des Patienten beendet. Treffen nach diesem Befehl Daten des Perimeters ein, ertönt ein akustisches Signal. Die Daten werden dann in die Datei abgelegt welche als erste geöffnet wird (Standard Verhalten von PeriData). Der Befehl 'CLOSE\_PAT' kann nur von einem Praxiscomputersystem verwendet werden. Im Standalone-Betrieb hat dieser Befehl keine Wirkung. Beim Befehl 'CLOSE\_APP' wird PeriData beendet. Der Befehl 'POPUP\_APP' stellt das Hauptfenster von PeriData in den Vordergrund.

### 3.2 Messages als Befehlsschnittstelle

PeriData kann über ein unsichtbares Fenster dieselben Befehle wie über die Kommandozeile entgegennehmen, dazu muss sich PeriData bereits im Speicher befinden. Wenn sich PeriData noch nicht im Speicher befindet, ist das Message Fenster auch noch nicht vorhanden, bzw wird mit dem API Aufruf 'FindWindow' auch nicht gefunden. Eine Nachricht an PeriData wird in drei Schritten gesendet:

- 1) Suchen des Nachrichten-Fensters mit der Windows-API Funktion 'FindWindow'.
- 2) Ermitteln der Nachrichten-Nummer mit der Windows-API Funktion 'RegisterWindowMessage'
- 3) Senden der Nachricht mit mit der Windows-API Funktion 'SendMessage'.

### 3.2.1 Messages

Im folgenden sind die Message erläutert, welche an PeriData gesendet werden können.

<b>Registrierter Message-String</b>	<b>Erläuterung</b>
MSG_CLOSE_APP	Beendet PeriData
MSG_POPUP_APP	Stellt PeriData in den Vordergrund
MSG_CLOSE_WND	Schliesst alle Fenster (keine Dialoge !)
MSG_CLOSE_PAT	Schliesst alle Fenster und beendet die Sitzung des Patienten

Für das Senden von Messages muss zuerst das Zielfenster von PeriData gesucht werden. Dieses Fenster besitzt den Titel (Caption) „PeriData Shell“ und hat den Schreibtisch als Vaterfenster (Desktop als Parent - 'Top-Level-Window'). Anschliessend kann eine registrierte Nachricht (ohne weitere Parameter) an PeriData uebergeben werden.