



Kassenärztliche  
Bundesvereinigung

Körperschaft des öffentlichen Rechts

## ***XPM-LDK.praxis***

*Handbuch Prüfmodul*

[KBV\_ITA\_AHEX\_Handbuch\_Pruefmodul  
\_LDK\_praxis]

Dezernat Digitalisierung und IT

10623 Berlin, Herbert-Lewin-Platz 2

Kassenärztliche Bundesvereinigung

Version 1.0.4  
Datum: 28.01.2024  
Klassifizierung: Extern  
Status: In Kraft

## DOKUMENTENHISTORIE

Version	Datum	Autor	Änderung	Begründung	Seite
1.0.4	28.01.2024	KBV	Zu verwendende minimale Java-Version aktualisiert		5
1.0.3	01.03.2018	KBV	Dateiname des Java-Archivs im Abschnitt 4.2 aktualisiert		22
1.0.2	03.08.2017	KBV	Einfügen der Javaklasse „XPMEinstieg“		22
1.0.1	15.06.2017	KBV	Anpassung der Beschreibung der Datumsprüfung Einfügung eines Hinweises bei der Beschreibung der Prüfungen		23 24
1.0.0	15.05.2017	KBV	Initiale Erstellung		all

## Inhaltsverzeichnis

<b>1. EINLEITUNG</b>	<b>5</b>
1.1 Begriffsklärung .....	5
1.2 Systemvoraussetzungen .....	6
<b>2. VERZEICHNISSTRUKTUR</b>	<b>7</b>
2.1 Ordner 'Bin' .....	7
2.2 Ordner 'Daten' .....	7
2.3 Ordner 'Doku' .....	7
2.4 Ordner 'Geprueft/Abgelehnt' .....	7
2.5 Ordner 'Geprueft/NichtAbgelehnt' .....	7
2.6 Ordner 'Konfig' .....	7
2.7 Ordner 'Listen' .....	7
2.8 Ordner 'Schema' .....	7
<b>3. ANWENDUNG DES PRÜFMODULS</b>	<b>8</b>
3.1 Technische Hinweise .....	8
3.2 Prüfmodi des Prüfmoduls .....	8
3.3 Konfigurationsdatei .....	9
3.3.1 Abschnitt: Allgemeiner Teil .....	9
3.3.2 Abschnitt: Eingabedateien .....	11
3.3.3 Abschnitt: Ausgabedateien .....	11
3.3.4 Abschnitt: Schalter .....	13
3.3.5 Umgebungsvariablen .....	14
3.4 Arbeiten im Kommandozeilenmodus .....	14
3.4.1 Übergabeparameter .....	15
3.4.2 Beispiele .....	16
3.5 Arbeiten mit grafischer Oberfläche .....	18
3.5.1 Prüfdatei selektieren .....	18
3.5.2 Prüfung starten .....	19
3.5.3 Optionen .....	20
3.5.4 Ansicht .....	20

## **4. HINWEISE ZU JAVA**

**21**

4.1	Starten eines Java-Programms .....	21
-----	------------------------------------	----

## **5. PRÜFUNGEN DES XPM-LDK**

**23**

5.1	Prüfungen bei Verwendung des Modus „Digitales Muster“ oder „Digitales Muster mit Verzeichnisprüfung“ .....	23
5.1.1	Einhaltung Anzahl erlaubter Zeichen .....	24
5.1.2	Einhaltung der vorgegebenen Schriftgrößen je Textfeld .....	24
5.1.3	Einhaltung des Formats bei Datumsfeldern .....	24
5.1.4	Für alle Textfelder ist die Option „Textlauf“ deaktiviert .....	24
5.1.5	Für alle Felder ist die Option „nur lesend“ gesetzt .....	24
5.1.6	Für alle Felder ist die Option „sichtbar“ gesetzt .....	24
5.1.7	Das Formular wird gegen die PDF/A-3a – Spezifikationen validiert .....	25
5.1.8	Es wird geprüft, ob alle Felder vorhanden und korrekt benannt sind .....	25
5.2	Prüfungen im Prüfmodus „Abgleich zwischen digitalem Muster und LDT“ .....	26
5.2.1	Übergreifend .....	26
5.2.2	Prüffälle für Muster 10 .....	31
5.2.3	Prüffälle für Muster 10A .....	33
5.3	Prüfungen im Prüfmodus „LDT“ bzw. „LDT Verzeichnisprüfung“ .....	34

## 1. Einleitung

Dieses Dokument gibt einen Überblick über die Arbeitsweise und eine genaue Beschreibung zur Installation und Ausführung des neuen LDK Prüfmoduls. Das Prüfmodul LDK ist ein XPM (Prüfmodul) der KBV. Daher wird „XPM-LDK“ und „LDK-Prüfmodul“ im vorliegenden Dokument synonym verwendet.

Das LDK-Prüfmodul ist ein Prüfprogramm für LDT-Datensätze und digitale Muster, die der definierten Schnittstellenbeschreibung LDT3.0 und in Bezug auf digitale Muster dem technischen Handbuch digitale Vordrucke entsprechen müssen. Das Prüfmodul kann in verschiedenen Prüfmodi ausgeführt werden. Diese Prüfmodi sind im Kapitel 3.2 des vorliegenden Handbuches beschrieben. Das Prüfmodul wird mit verschiedenen Startskripten und Übergabeparametern ausgeliefert, mit denen die verschiedenen Prüfmodi abgebildet sind.

Als Ergebnis der Prüfung wird unter anderem ein Prüfprotokoll erstellt, das Informationen über den Zustand der Prüfdatei enthält. Als Ergebnis der Prüfung sind je nach Zustand der Prüfdatei folgende Statusmeldungen möglich: Ok, Warnung, Fehlerhaft, Hinweis oder Abbruch.

Das LDK-Prüfmodul ist auf allen Computersystemen lauffähig, für die die Java Laufzeitumgebung in der Version 17 oder höher verfügbar ist.

### 1.1 Begriffsklärung

- **XPM\_LDK:** LDK steht für Labordatenkommunikation. Unter diesem Begriff sind der LDT3.0 und digitale Muster 10 bzw. 10A zusammengefasst. Das XPM\_LDK steht als Synonym für das Prüfmodul der digitalen Muster 10 und 10 A sowie für den LDT 3.0 Datensatz.
- **XPM\_LDK-Paket:** Ein Paket des LDK-Prüfmoduls bestehend aus folgenden Teilen :
  - XPM-Kernel (JAVA-Programm zum Interpretieren des LDK-Prüfprojektes)
  - LDK-Prüfprojekt (LDK spezifische Erweiterung des XPM-Kernels)
  - ReleaseNotes (Textdatei, die alle Informationen zur Version des Prüfmodulpakets enthält)
  - GUI (Aufruf des Prüfmoduls als graphische Benutzungsoberfläche)
  - Batchdateien und Shellskripte (Beispiele für den kommandozeilenorientierten Aufruf in verschiedenen Modi)
- **XPM-LDK.praxis-Va.b.c:** Schnittstellenspezifisches Prüfmodulpaket der Schnittstelle LDT3.0 und für digitale Muster. Prüfmodul-Version a.b.c auch "das aktuelle LDK-Prüfmodul" genannt.
- **XPM\_LDK-Prüfprojekt:** Tabellen und Codeelemente, die die schnittstellenspezifische Konfiguration enthalten; schnittstellenspezifischer Bestandteil des XPM\_LDK - Pakets.
- **XPM-Kernel:** Programm zum Interpretieren des XPM\_LDK-Prüfprojektes, der Kernel ist ohne verfahrensspezifisches Prüfprojekt nicht verwendbar; XPM-allgemeiner Bestandteil des XPM\_LDK-Pakets.
- **"Prüfmodul-Version":** Bezeichnet die Version des XPM\_LDK-Paketes.

- **"das aktuelle LDK-Prüfmodul"**: Hiermit ist meist die neuste Version des XPM\_LDK-Pakets gemeint.
- **"Gesamtpaket XPM-LDK.praxis-Va.b.c"**: Das Gesamtpaket enthält alle relevanten Dokumente und Software für die LDT-Schnittstelle und die digitalen Muster. Achtung! Die Versionsnummer des Gesamtpakets ist nicht zu verwechseln mit der Versionsnummer des technischen Handbuchs oder des Prüfmoduls.

## 1.2 Systemvoraussetzungen

XPM ist eine Applikation für ein 32bit-System.

Benötigt wird ein Computersystem, das leistungsmäßig mit einem IBM-kompatiblen PC, 256 MB Hauptspeicher und einem Pentium-Prozessor 500 oder höher vergleichbar ist.

Langsamere Prozessoren und weniger Hauptspeicher erhöhen stark die Laufzeit des Programms. XPM benötigt weniger als 10 MByte Festplattenplatz.

## 2. Verzeichnisstruktur

Die Verzeichnisstruktur des LDK-Prüfmoduls hat folgenden Aufbau:

### 2.1 Ordner 'Bin'

Dieser Ordner beinhaltet alle Java-Archive und binären Steuerdateien, die zur Ausführung des LDK-Pakets benötigt werden.

### 2.2 Ordner 'Daten'

Dieser Ordner enthält die Prüfdateien.

### 2.3 Ordner 'Doku'

Dieser Ordner enthält die Dokumentation zum XPM\_LDK.

### 2.4 Ordner 'Geprueft/Abgelehnt'

Dieser Ordner dient als Ablage für geprüfte Dateien. Hier werden die Dateien abgelegt, die fehlerhaft sind.

### 2.5 Ordner 'Geprueft/NichtAbgelehnt'

Dieser Ordner dient als Ablage für geprüfte Dateien. Hier werden die Dateien abgelegt, die fehlerfrei sind.

### 2.6 Ordner 'Konfig'

Dieser Ordner enthält Konfigurationsdatei(en) im XML-Format.

### 2.7 Ordner 'Listen'

In diesem Ordner werden alle Ausgaben des Prüfmoduls generiert.

### 2.8 Ordner 'Schema'

In diesem Ordner befinden sich die XML-Schemadateien, die das Prüfformat definieren. Die Schemadateien (\*.xsd) dürfen **nicht** verändert werden.

### 3. Anwendung des Prüfmoduls

XPM\_LDK steht als einheitliches Werkzeug zur Prüfung möglichst vieler Eingangs- und Ausgangsdaten zur Verfügung. Dabei verarbeitet es je nach gewähltem Startskript LDT-Dateien sowie digitale Muster. XPM\_LDK läuft in einer Java Laufzeitumgebung und kann somit auf allen Betriebssystemen eingesetzt werden, auf denen die Java-Laufzeitumgebung installiert ist.

Ein Prüfmodullauf beinhaltet dabei den Start des Moduls über eines der mitgelieferten Startskripts (.bat) oder über den Kommandozeilenaufruf mit entsprechendem Parameter. Danach werden die dem Prüfmodul übergebenen Dateien geprüft und ein Ergebnis ausgegeben. Die Konfiguration eines Prüfmodullaufs erfolgt mit Hilfe einer XML-Konfigurationsdatei. Nähere Informationen zur Konfigurationsdatei finden Sie in Abschnitt 3.3.

#### 3.1 Technische Hinweise

XPM\_LDK besteht im Kern aus einem Steuermodul (pruefprogramm.ldk.x.x) sowie einem Prüfungsmodul (pruefungLDK.jar).

Zur Prüfung des Eingangsformates von LDT3.0-Dateien benötigt XPM\_LDK Informationen über den Dokumentenaufbau. Diese Informationen erhält XPM\_LDK auch zur Laufzeit durch Auswertung (Interpretation) des entsprechenden XML-Schemas. Digitale Muster werden nicht gegen ein Schema geprüft sondern gegen ein internes Regelwerk.

Das Ergebnis einer Prüfung (Fehlermeldungen, Informationsmeldungen) sowie evtl. andere Ausgabedateien (Statistikmeldungen) werden in Dateien ausgegeben, deren Format vom Benutzer konfigurierbar ist.

#### 3.2 Prüfmodi des Prüfmoduls

Das Prüfmodul LDK wird, wie eingangs erwähnt, mit verschiedenen Übergabeparametern und Startskripten ausgeliefert. Die folgende Tabelle soll Aufschluss darüber geben, wie diese zusammenhängen. Die genaue Funktion der Übergabeparameter wird in Kapitel 0 erläutert.

Prüfmodus	Übergabeparameter	Skriptname
LDT	-c, -f Optional: -h, -m, -p, -s, -v, -z, -e,	StartPruefung.bat und StartPruefung .sh
LDT Verzeichnisprüfung	-c, -e, -s, -m, Optional: -f, -h, -p, -v, -z	VerzeichnisPruefung.bat und VerzeichnisPruefung.sh



Digitales Muster	-c, -x, -f Optional: -e, -s, -m, -v, -h	StartPruefungDigitaleMuster.bat und StartPruefungDigitaleMuster.sh
Digitales Muster mit Verzeichnisprüfung	-s, -x, -e, -m, -c Optional: -h, -v	VerzeichnisPruefungDigitaleMuster.bat und VerzeichnisPruefungDigitaleMuster.sh
Abgleich zwischen digitalem Muster und LDT	-c, -y, -f	StartPruefungDigitaleMusterVsLDT.bat und StartPruefungDigitaleMusterVsLDT.sh

### 3.3 Konfigurationsdatei

Neben den im Abschnitt (3.2) genannten Übergabeparametern und mitgelieferten Startskripten können in der Konfigurationsdatei übergreifende Einstellungen vorgenommen werden.

In diesem Abschnitt folgt eine allgemeine Einführung in den Aufbau einer Konfigurationsdatei. In den jeweiligen Konfigurationsdateien selbst existiert zu jedem Konfigurationselement eine spezielle Beschreibung.

Die Pfadangaben in der Konfigurationsdatei müssen eventuell dem jeweiligen Betriebssystem angepasst werden. Die Konfigurationsdateien im Lieferumfang sind so voreingestellt, dass keinerlei Anpassungen nötig sind. Als Trennzeichen für Verzeichnisse wird das Zeichen '/' verwendet. Diese Voreinstellung erlaubt die Nutzung gleicher Konfigurationsdateien auf verschiedenen Betriebssystemen (Windows, Unix, Linux, ...). Relative Pfadangaben werden als relativ zum Installationsverzeichnis betrachtet. Die Konfigurationsdatei wird in 4 Abschnitte eingeteilt.

#### 3.3.1 Abschnitt: Allgemeiner Teil

Im allgemeinen Teil werden allgemeine Informationen zum Prüflauf eingestellt.

##### 3.3.1.1 Installationsverzeichnis

Das Installationsverzeichnis wird im Element „**pruefppfad**“ festgelegt.

z.B.: `<pruefppfad>./</pruefppfad>`

##### 3.3.1.2 Prüfdatenverzeichnis

Das Verzeichnis mit Prüfdateien wird im Element „**pruefdaten**“ festgelegt.

z.B.: `<pruefdaten>Daten/</pruefdaten>`

##### 3.3.1.3 Geprüft-OK-Verzeichnis

Bei eingeschalteter Option `-m` wird die geprüfte Datei mit Status 'ok' in dieses Verzeichnis kopiert. Pfadangabe erfolgt im Element „**okdaten**“.

z.B.: `<okdaten>Geprueft/NichtAbgelehnt/</okdaten>`

### 3.3.1.4 Ausschlussverzeichnis

Bei eingeschalteter Option `–m` wird die geprüfte Datei mit Status ‘abgelehnt’ bzw. ‘abbruch’ in dieses Verzeichnis kopiert. Pfadangabe erfolgt im Element „**fehlerdaten**“.

z.B.: `<fehlerdaten>Geprueft/Abgelehnt/</fehlerdaten>`

### 3.3.1.5 Protokolldatei

Bei jedem Prüflauf wird eine Protokolldatei geschrieben. Der Pfad der Protokolldatei kann in der Konfigurationsdatei im Element „**log\_datei**“ angegeben werden. Bei einem eventuellen fehlerhaften Programmablauf befinden sich in dieser Datei Informationen zur genauen Fehlerursache.

z.B.: `<log_datei>Listen/XPM_Logfile.log</log_datei>`

### 3.3.1.6 Dateifilter

Die Menge der zu prüfenden Dateien kann über das Element „**datei\_filter**“ eingegrenzt werden.

z.B.: `<datei_filter>*.LDT</datei_filter>`

### 3.3.1.7 PDF-Dateifilter

Die Menge der zu prüfenden PDF-Dateien kann über das Element „**pdf\_datei\_filter**“ eingegrenzt werden.

z.B.: `<pdf_datei_filter>*.PDF</pdf_datei_filter>`

### 3.3.1.8 Datensatzbeschreibung

Die Datensatzbeschreibung LDT3.0 wird in Form einer XML-Schemadatei spezifiziert.

Die Pfadangabe zur XML-Schemadatei wird im Element „**pruefschema**“ festgehalten.

z.B.: `<pruefschema>Schema/ldtdata.xsd</pruefschema>`

**Hinweis:** Sollte in der Pfadangabe des Schemas das ‘%’-Zeichen verwendet werden, so muss dieses durch die Zeichenkette ‘%25’ ersetzt werden.

### 3.3.1.9 Warnungen

Diese Einstellung ermöglicht es, das Protokollieren der Warnungen ein- bzw. auszu-schalten. Die Konfiguration wird im Element „**warnungen**“ festgelegt.

Wertebereich: „ja“, „nein“.

z.B.: `<warnungen>ja</warnungen>`

### 3.3.1.10 Begrenzung der Fehlermeldungen

Standardmäßig begrenzt das XPM\_LDK die gemeldeten Fehler einer bestimmten Meldungsnummer auf eine festgelegte maximale Anzahl. Die maximale Anzahl kann je nach Prüfprojekt variieren und beträgt in der Regel zwischen 30 und 50. Diese Einstellung dient der Übersichtlichkeit eines Fehlerprotokolls, um die Liste nicht mit systematischen Fehlern zu überladen. Die standardmäßige Begrenzung kann ein- und ausgeschaltet werden.

Wertebereich: „ja“, „nein“.

z.B.: `<fehler_begrenzen>ja</fehler_begrenzen>`

### 3.3.1.11 Dokumentation

Mit dieser Einstellung kann auf eine alternative Dokumentationsdatei verwiesen werden, welche in der GUI unter „Hilfe / Hilfe“ verlinkt ist.

z.B.: `<dokumentation>Doku/Handbuch_XPM_LDK.pdf</dokumentation>`

### 3.3.2 Abschnitt: Eingabedateien

Im Abschnitt Eingabedateien wird der Pfad zur Steuertabelle festgelegt. Die Steuertabelle für das XPM\_LDK, im Element „kbv\_tabelle“ spezifiziert, muss in der Konfigurationsdatei angegeben werden. In der Regel ist an dieser Steuertabelle durch den Anwender kein Eingriff notwendig.

z.B.: `<kbv_tabelle> Bin/ldk_tabelle.bin</kbv_tabelle>`

### 3.3.3 Abschnitt: Ausgabedateien

Im Abschnitt Ausgabedateien werden die Pfade für die Ausgabelisten und Protokolle festgelegt.

Über das Attribut Format wird das Ausgabeformat festgelegt:

- |                  |   |
|------------------|---|
| ▪ CSV            | Kommaseparierte Ausgabe, über das Attribut 'Trennzeichen' lässt sich das Trennzeichen zwischen den Spalten festlegen. Standardmäßig ist hier das Komma voreingestellt.  |
| ▪ HTML           | HTML-Format   |
| ▪ JRPRINT        | Internes Ausgabeformat, kann vom Prüfmodul angezeigt und gedruckt werden  |
| ▪ PDF            | Portable Document Format  |
| ▪ PRINTER        | Direktausgabe auf den Drucker   |
| ▪ PRINTER_DIALOG | Direktausgabe auf den Drucker mit Einstellungsfenster   |
| ▪ RTF            | Rich Text Format, formatiertes Textformat   |
| ▪ TEXT           | ASCII Text, über das Attribut 'Seitenbreite' lässt sich die Seitenbreite in Zeichen festlegen. Standardmäßig ist hier die Breite von 80 Zeichen voreingestellt. Ein ansprechendes Layout erreicht man, in dem die Breite auf den Wert 120 gesetzt wird. |
| ▪ XLS            | Microsoft Excel-Format  |
| ▪ XML            | XML-Format  |

Tabelle 1: Ausgabeformate

z.B.: <FehlerListe Format="PDF">Listen/Protokoll.pdf</FehlerListe>

PDF Dateien können angezeigt und gedruckt werden.

Alles, was dazu benötigt wird, ist der Adobe Reader®, der kostenlos unter <https://get.adobe.com/de/reader/> heruntergeladen werden kann. XPM\_LDK erzeugt PDF Dokumente, die vom Acrobat Reader® ab der Version 5.0 und höher angezeigt werden können.

Beim Verarbeiten von Massendaten (Servermodus, zip-Archive) werden die Ausgabedateien vom Prüfmodul eigenständig umbenannt. Der Name der Ausgabedatei setzt sich zusammen aus dem Namen der Prüfdatei und den vorgegebenen Dateinamen. Diese Vorgehensweise verhindert das Überschreiben bereits erzeugter Protokolle.

Bei Fehlerprotokollen fügt das XPM noch ein Präfix hinzu, das den Errorlevel der Prüfung kennzeichnet. Es werden folgende Präfixe verwendet:

Errorlevel	Präfix
0	Ok_
1	Warnung_
2	Fehler_
3	Abbruch_

Tabelle 2: ErrorLevel

So wird bei einer Prüfdatei mit dem Namen 'Test.xml' und dem Errorlevel 1 der Name „Warnung\_Test.xml.Fehler.pdf“ für die Fehlerliste vergeben.

Zusätzlich wurde eine zweite flexiblere Umbenennungsmethode integriert.

Sobald im Namen einer Ausgabedatei die Variable `${DATEI_NAME}` verwendet wird, ersetzt XPM diese Variable durch den Namen der Prüfdatei. Fehlt die Variable `${DATEI_NAME}`, dann wird der Name der Prüfdatei als erstes im Namen der Ausgabedatei eingefügt.

Bei der Umbenennung von Fehlerprotokollen kann man mit der Variablen `${STATUS}` den Fehlerstatus an einer beliebigen Stelle im Dateinamen platzieren. Fehlt die Variable `${STATUS}`, dann wird der Fehlerstatus als erstes im Namen der Ausgabedatei eingefügt.

### 3.3.3.1 Statistikdatei

Im Abschnitt Ausgabedateien Element '**Fehlerstatistik**' wird der Pfad für eine Statistikliste festgelegt.

z.B.:

<StatistikListe Format="PDF">./Listen/FehlerStatistik.pdf</StatistikListe>

In der Statistikliste wird für jede Prüfdatei, für die eine Meldung des Prüfmoduls erfolgte, eine Meldungsstatistik ausgegeben. Die mit dem Status 'ok' geprüften Dateien tauchen in der Statistikliste nicht auf.

Die auftretenden Meldungstexte können das Zeichen '%s' enthalten. Dies ist kein Programmfehler sondern nur ein Hinweis darauf, dass diese Meldung variable Inhalte ent-

hält, die erst zur Laufzeit ermittelt werden und unterschiedliche Ausprägungen enthalten kann.

z.B.: Zu der GNR '%s' wurde keine Angabe im Feld '%s' gemacht.

### 3.3.4 Abschnitt: Schalter

Über die Konfigurationsschalter kann der Programmablauf modifiziert werden.

Folgende Schalter sind definiert:

#### 3.3.4.1 Schalter: Regelwerksprüfung für LDT-Dateien

Hier werden die im Rahmen der Prüfmodi für „LDT“ zu prüfenden Regelwerke definiert:

- **basis:**

Es werden nur die Regeln geprüft, die in der Regeltabelle des LDT 3.0 mit „Basis“ gekennzeichnet sind

- **kbv:**

Es werden nur die Regeln geprüft, die in der Regeltabelle des LDT 3.0 mit „Basis“ und mit „KBV“ gekennzeichnet sind

- **qms\*:**

Es werden nur die Regeln geprüft, die in der Regeltabelle des LDT 3.0 mit „Basis“ und mit „QMS“ gekennzeichnet sind

- **gesamt\*:**

Es werden alle Regeln geprüft.

Wertebereich: „basis“, „kbv“, „qms\*“, „gesamt\*“

z.B.: `<pruef_modus>gesamt</pruef_modus>`

\* *nur verfügbar nach erfolgreicher Zertifizierung des den LDT-Datensatz erstellenden Systems durch die Zertifizierungsstelle des QMS e.V.*

#### 3.3.4.2 Schalter: Änderung der Konfigurationsdatei über die GUI

Mit dieser Einstellung kann die Möglichkeit der Änderung der Konfigurationsdatei über die GUI zugelassen bzw. nicht zugelassen werden.

Wertebereich: „ja“, „nein“

z.B.: `<gui_optionen>ja</gui_optionen>`

#### 3.3.4.3 Schalter: Änderung der Prüfdatei über die GUI

Mit dieser Einstellung kann die Möglichkeit der Änderung der Prüfdatei über die GUI zugelassen bzw. nicht zugelassen werden.

Wertebereich: „ja“, „nein“

z.B.: `<gui_selektion>nein</gui_selektion>`

### 3.3.5 Umgebungsvariablen

Jedes Element der Konfigurationsdatei darf Umgebungsvariablen enthalten.

Diese Umgebungsvariablen müssen der JavaVM jedoch über den Übergabeparameter `-D` übergeben werden. Nach dem Einlesen der Konfigurationsdatei werden die Umgebungsvariablen durch ihre Werte ersetzt. Findet das XPM\_LDK eine Umgebungsvariable nicht, wird der Prüflauf abgebrochen.

Mit Hilfe von Umgebungsvariablen kann mehreren Benutzern eine separate Umgebung zur Verfügung gestellt werden, die auf eine einzige Installation zugreifen.

Beispiel:

In der Konfigurationsdatei wird der Prüfpfad folgendermaßen festgelegt:

```
<pruefpfad>%INSTALLATION%/%UMGEBUNG%/</pruefpfad>
```

Die zwei Umgebungsvariablen `INSTALLATION` und `UMGEBUNG` müssen entweder in einer Batchdatei bzw. einem Shellskript:

```
set INSTALLATION=C:\Projekte\JavaPruefmodul\Test
set UMGEBUNG=LDT.Praxis
```

oder in der aufrufenden Applikation entsprechend gesetzt werden.

Jetzt muss nur noch dafür gesorgt werden, dass die Umgebungsvariablen der JavaVM bekannt sind.

Über den folgenden Aufruf werden die Umgebungsvariablen unter gleichem Namen dem XPM\_LDK bekannt gegeben.

```
java -DINSTALLATION=%INSTALLATION% -DUMGEBUNG=%UMGEBUNG% ...
```

### 3.4 Arbeiten im Kommandozeilenmodus

Das Prüfmodul kann als ein kommandozeilenorientiertes Programm gestartet werden. Der Lauf des Prüfmoduls wird mithilfe von Übergabeparametern gesteuert. Es folgt eine Auflistung aller Übergabeparameter.

### 3.4.1 Übergabeparameter

Übergabeparameter	Beschreibung
-c	Das Prüfmodul braucht für die Prüfung die Pfadangabe einer XML-Konfigurationsdatei. Hinter dieser Option muss die Pfadangabe stehen!
-e	Das Prüfmodul wird im Einzellaufmodus gestartet. XPM_LDK verarbeitet alle Dateien eines Eingangsverzeichnis und beendet sich anschließend. Diese Option ist nur in Kombination mit Servermodus aufrufbar. Dieser Übergabeparameter ist optional.
-f	Hinter dieser Option sollte die Pfadangabe einer Prüfdatei stehen, die vom Prüfmodul bearbeitet wird. Dieser Übergabeparameter ist optional.
-h	Das Prüfmodul gibt einen Hilfetext aus und beendet sich anschließend.
-m	Das Prüfmodul verschiebt bereits geprüfte Dateien bzw. Zip-Archive in entsprechende Verzeichnisse, die in der Konfigurationsdatei eingestellt werden. Dieser Übergabeparameter ist optional.
-p	Alle Konfigurationsmöglichkeiten der Konfigurationsdatei, können nun über diesen Übergabeparameter gesetzt werden. Die Zuweisung erfolgt in der Form 'Schalter[@Attribut]=Wert'. Schalter ist ein beliebiger Schalter der Konfigurationsdatei. Soll nur das Attribut eines Schalters gesetzt werden so muss der Name des Attributes hinter dem @-Zeichen angegeben werden. Der Wert selbst wird hinter dem Gleichheitszeichen angegeben. Dieser Parameter kann mehrfach übergeben werden, um diverse Einstellungen vorzunehmen. Bitte beachten Sie, dass die hier übergebenen Parameter die Einstellungen der Konfigurationsdatei überschreiben. Schauen Sie sich zum besseren Verständnis das <a href="#">Beispiel 4</a> an. Dieser Übergabeparameter ist optional.
-s	Das Prüfmodul wird im sogenannten Servermodus gestartet. XPM_LDK verarbeitet im 30 Sekunden-Takt Dateien eines Eingangsverzeichnis. Der Abbruch des Programmlaufs kann über CTRL-C erfolgen. Dieser Übergabeparameter ist optional.

-v	Das Prüfmodul gibt die Versionsnummer des XPM-Kernels aus und beendet sich anschließend. Wird außer diesem Übergabeparameter auch noch die Konfigurationsdatei (Übergabeparameter -c) angegeben, so wird zusätzlich die Versionsnummer des Prüfpaketes ausgegeben.
-z	Hinter dieser Option sollte die Pfadangabe einer Zip-Datei stehen, die vom Prüfmodul bearbeitet wird. Der Inhalt des Zip-Archivs darf nur Prüfdateien enthalten. Dieser Übergabeparameter ist optional.
-x	Der Parameter -x aktiviert die Prüfung der digitalen Muster. Der Pfad zum PDF wird mit -f %PFAD% übergeben, außer bei der Verzeichnisprüfung. Der Parameter darf nicht gleichzeitig mit -y aufgerufen werden. Bsp: -x -f ./DigitaleMuster
-y	Der Parameter -y aktiviert den PDF-LDT abgleich zwischen der PDF-Datei, deren Pfad hinter diesem Parameter angegeben wird und der LDT-Datei, deren Pfad hinter dem Parameter -f angegeben wird. Dieser Parameter darf nicht gleichzeitig mit dem Parameter -x aufgerufen werden. Bsp: -y ./DigitaleMuster/xyz.pdf -f ./LDT/Z0112345.ldt

Tabelle 3: Übergabeparameter

### 3.4.2 Beispiele

Nun folgen Beispiele für den Aufruf des XPM\_LDK.

#### 3.2.2.1. Beispiel 1: Prüfen einer einzelnen Datei

Übergabeparameter:

```
-c Konfig/konfig.xml -f Daten/Z0112345.ldt
```

XPM\_LDK liest die Konfigurationsdatei 'Konfig/konfig.xml' ein und prüft die Datei 'Daten/Z0112345.LDT'. Anschließend wird XPM\_LDK beendet.

#### 3.2.2.2. Beispiel 2: Prüfen eines Verzeichnisses

Übergabeparameter:

```
-c Konfig/konfig.xml -s -e
```



XPM\_LDK liest die Konfigurationsdatei 'Konfig/konfig.xml' ein und prüft das komplette Verzeichnis, welches in der Konfigurationsdatei unter dem Konfigurationsschalter 'pruefdaten' angegeben wurde. Anschließend wird XPM\_LDK beendet.

### 3.2.2.3. Beispiel 3: Prüfen eines Verzeichnisses (ohne automatisches Beenden)

Übergabeparameter:

```
-c Konfig/konfig.xml -s
```

XPM\_LDK liest die Konfigurationsdatei 'Konfig/konfig.xml' ein und prüft das komplette Verzeichnis, welches in der Konfigurationsdatei unter dem Konfigurationsschalter 'pruefdaten' angegeben wurde. Alle 30 Sekunden wird das angegebene Verzeichnis abgeprüft.

### 3.2.2.4. Beispiel 4: Setzen des Pfades und des Formates einer Ausgabeliste

Übergabeparameter:

```
-p FehlerListe=Listen/Protokoll.xml -p FehlerListe@Format=XML
```

Die Ausgabeliste mit dem Namen 'FehlerListe' bekommt den Pfad 'Listen/Protokoll.xml' zugewiesen. Das Format der Ausgabeliste wird auf PDF gesetzt.

### 3.2.2.5. Beispiel 5: Prüfen einer PDF – Datei

Übergabeparameter:

```
-c Konfig\konfig.xml -x -f Daten\Muster_10.pdf
```

XPM\_LDK liest die Konfigurationsdatei „konfig.xml“ ein und prüft das PDF, welches über den Pfad hinter -f angegeben wird.

### 3.2.2.6. Beispiel 5: Vergleich einer PDF – Datei mit einer LDT Datei

```
-c Konfig\konfig.xml -f Daten\Z01Auftrag.ldt -y Daten\Muster_10.pdf
```

Das XPM\_LDK liest die Konfigurationsdatei „konfig.xml“ ein und vergleicht den Inhalt des PDF's mit dem Inhalt der LDT-Datei.

### 3.5 Arbeiten mit grafischer Oberfläche

Die grafische Oberfläche stellt eine Alternative zum Kommandozeilen-Programm dar. Es können sowohl Verzeichnisse als auch Zip-Archive geprüft werden.

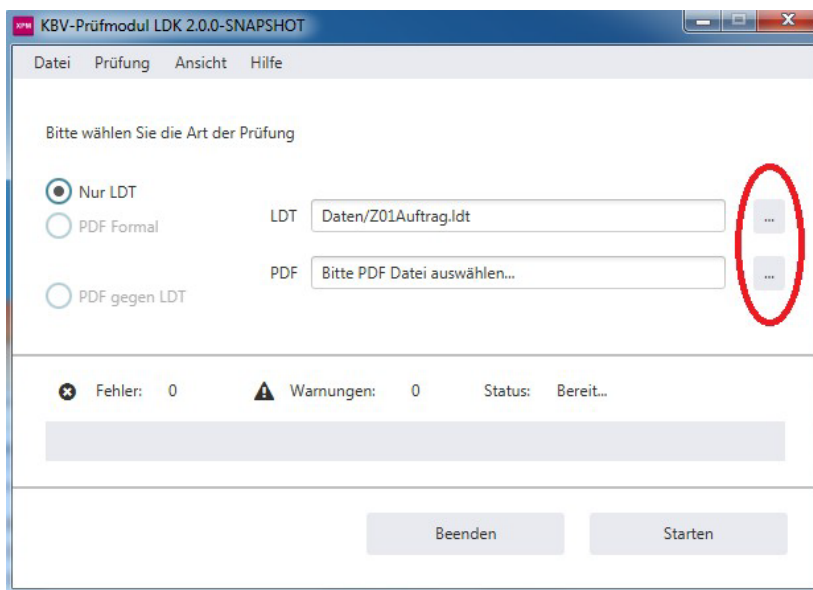
#### 3.5.1 Prüfdatei selektieren

Bitte drücken Sie den jeweiligen ‘...’-Knopf und wählen Sie die gewünschte Prüfdatei aus.

Die Modi „PDF Formal“ und „PDF gegen LDT“ sind auswählbar, sobald eine PDF-Datei und eine LDT-Datei ausgewählt wurde.

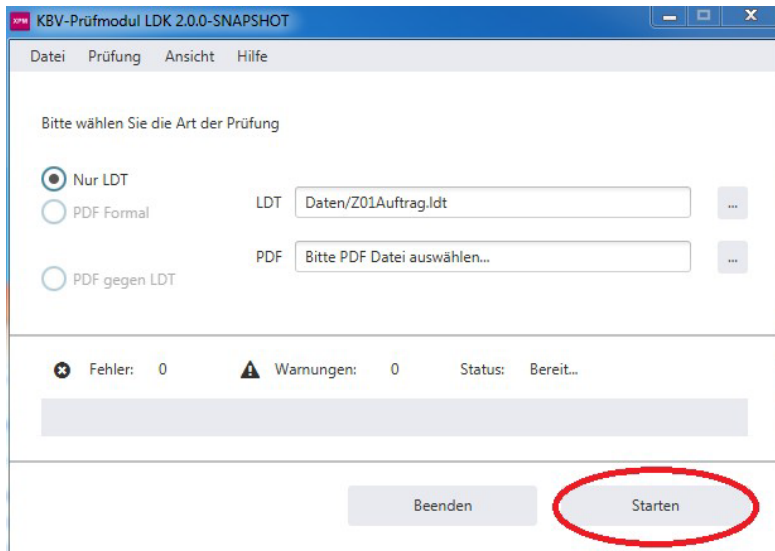
Ist lediglich eine PDF-Datei ausgewählt, kann nur der Modus „PDF Formal“ verwendet werden.

Ist lediglich eine LDT-Datei ausgewählt, kann nur der Modus „Nur LDT“ ausgewählt werden.



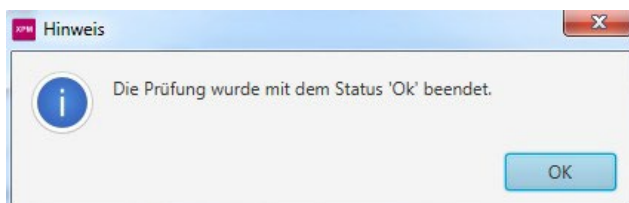
### 3.5.2 Prüfung starten

Bitte drücken Sie den 'Starten'-Knopf.

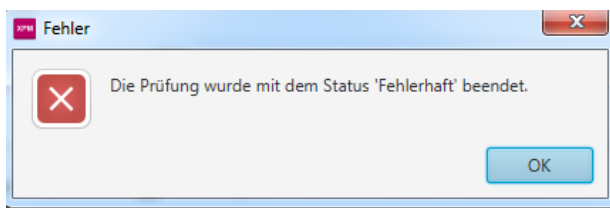


Während der Prüfung zeigt der Fortschrittsbalken den aktuellen Fortschritt.

Nach erfolgter Prüfung erscheint eine Meldung mit dem entsprechenden Prüfstatus:

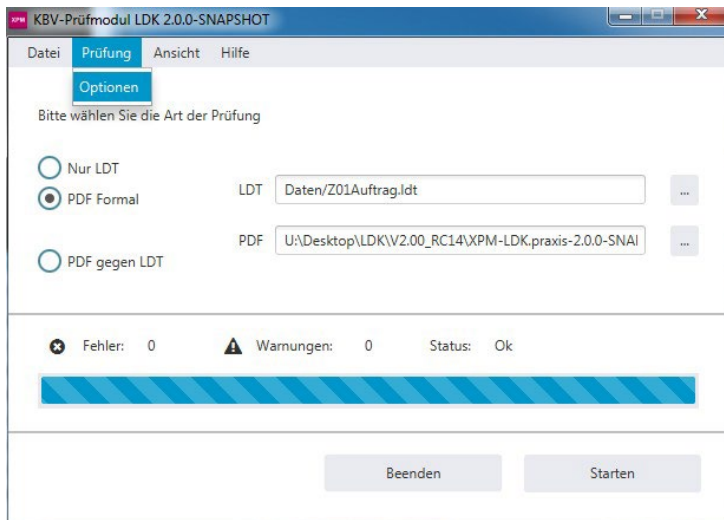


oder



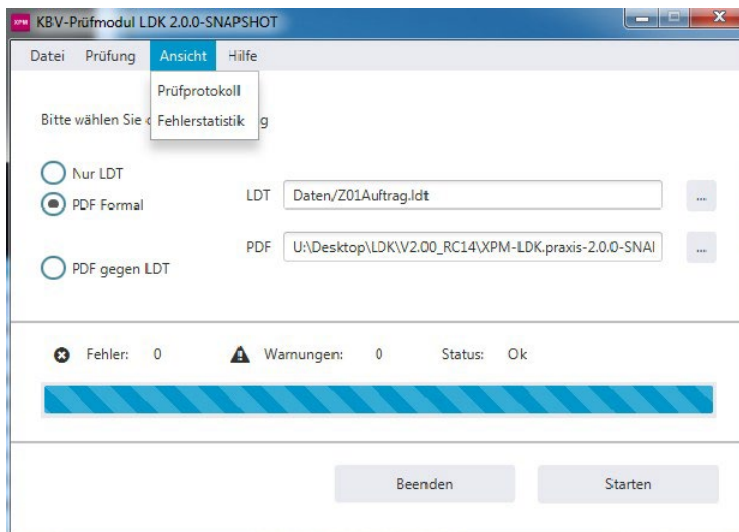
### 3.5.3 Optionen

Unter den Optionen kann das Prüfmodul konfiguriert werden.



### 3.5.4 Ansicht

Unter „Ansicht“ kann die Fehlerstatistik und das Prüfprotokoll aufgerufen werden.



## 4. Hinweise zu Java

### 4.1 Starten eines Java-Programms

Das LDK-Prüfmodul ist eine Java-Applikation und wird in einer Java Laufzeitumgebung ausgeführt.

Hier ein Beispiel für einen Aufruf:

```
java -Xmx300m
    -Dfile.encoding=8859_1
    -cp "Bin/*"
    de.kbv.pruefmodul.GUI.LDK
    -c Konfig/konfig.xml -f Daten/Z011312345.ldt
```

Der Befehl 'java' startet die virtuelle Maschine von Java.

Der Parameter '-Xmx300m' erlaubt der Java Laufzeitumgebung einen Hauptspeicher von bis zu 300 MB zu reservieren. Diese Option garantiert einen stabilen Programmlauf bei Abrechnungsdateien in der Größenordnung bis ca. 300 MB.

Der Parameter '-Dfile.encoding=8859\_15' stellt den entsprechenden Zeichensatz ein und ermöglicht hier die Verwendung von deutschen Umlauten.

Der Parameter '-cp "Bin/\*"' spezifiziert alle Java-Archive, die für den Programmablauf benötigt werden.

Der Parameter 'de.kbv.pruefmodul.GUI.LDK' ist der Name einer Klasse, die das XPM startet.

Die Parameter '-c' und '-f' sind die eigentlichen Übergabeparameter, die an das Prüfprogramm übergeben werden.

## 4.2 Starten des Prüfmoduls aus einem Java-Programm

Das LDK-Prüfmodul ist eine Java-Applikation und kann von einem anderen Java-Programm aufgerufen werden. Für eine leichtere Anbindung wurde die Klasse *de.kbv.pruefmodul.core.extern.XPMEinstieg* implementiert. Diese Klasse ist im Java-Archiv **xpm-core-<Versionsnummer>.jar** im Quellcode enthalten. Bitte schauen Sie sich hierzu die Methode 'main' etwas genauer an.

Beispielaufruf:

```
package de.org.beispiel

import de.kbv.pruefmodul.core.extern.XPMEinstieg;
import de.kbv.pruefmodul.modul.ldk.XPMAdapter;

public class XpmEinstiegTest {

    public static void main(String[] args) throws XPMEException {
        // TODO Auto-generated method stub
        System.out.println("user.dir: " + System.getProperty("user.dir"));
        XPMEinstieg xpm = new XPMEinstieg("src/test/resources/Konfig/konfig.xml",
                                         "src/test/resources/Daten/Z01Auftrag.ldt", new XPMAdapter());

        xpm.setServer(false);
        xpm.setZipFile(false);
        int nStatus = xpm.pruefe();
        System.out.println("Einzel-Prüfung mit Status " + nStatus + " beendet.");
        System.out.println();
    }
}
```

## 5. Prüfungen des XPM-LDK

In diesem Abschnitt werden die Prüfregeln aufgeführt, die das Prüfmodul bei der Analyse eines digitalen Musters prüft.

### 5.1 Prüfungen bei Verwendung des Modus „Digitales Muster“ oder „Digitales Muster mit Verzeichnisprüfung“

Das Prüfmodul liest die Formularnummer aus den Metadaten der zu prüfenden PDF-Datei aus. Wird eine Formularnummer ausgelesen, die nicht „10“ oder „10A“ entspricht, gibt das Prüfmodul dies als Fehlermeldung zurück und wird anschließend beendet.

Liegt ein Muster 10 oder 10A vor, führt das Prüfmodul nachfolgend genannte Prüffälle aus. Ziel dieser Prüfungen ist die technische Interoperabilität zwischen Sender und Empfänger sicherzustellen.

Hinweis:

Beim Auslesen der Formularnummer des digitalen Musters aus den Metadaten unterstützt das Prüfmodul sowohl die Darstellung der Metadaten als eigene Elemente als auch als Attribute (Details siehe Kapitel 7.9.2.2 der XPM-Spezifikation). Somit werden die folgenden beiden Darstellungen unterstützt:

**Darstellung mit Elementen:**

```
<ftx:ControlData rdf:parseType="Resource">
  <control:Anzahl_Zeichen_Titel>0</control:Anzahl_Zeichen_Titel>
  <control:Anzahl_Zeichen_Vorname>0</control:Anzahl_Zeichen_Vorname>
  <control:Anzahl_Zeichen_Namenszusatz>0</control:Anzahl_Zeichen_Namenszusatz>
  <control:Anzahl_Zeichen_Hausnummer>0</control:Anzahl_Zeichen_Hausnummer>
  <control:Anzahl_Zeichen_Postleitzahl>0</control:Anzahl_Zeichen_Postleitzahl>
  <control:Anzahl_Zeichen_Wohnsitzlaendercode>0</control:Anzahl_Zeichen_Wohnsitzlaendercode>
  <control:Auftragsnummer_Einsender>0</control:Auftragsnummer_Einsender>
  <control:Formularnummer>10</control:Formularnummer>
  <control:Formularversion>07.2017</control:Formularversion>
</ftx:ControlData>
```

**Darstellung mit Attributen:**

```
<ftx:ControlData
  control:Anzahl_Zeichen_Titel="8"
  control:Anzahl_Zeichen_Vorname="14"
  control:Anzahl_Zeichen_Namenszusatz="13"
  control:Anzahl_Zeichen_Hausnummer="3"
  control:Anzahl_Zeichen_Postleitzahl="5"
  control:Anzahl_Zeichen_Wohnsitzlaendercode="1"
  control:Auftragsnummer_Einsender="0"
  control:Formularnummer="10"
  control:Formularversion="07.2017"/>
```

### 5.1.1 Einhaltung Anzahl erlaubter Zeichen

Das Prüfmodul prüft für alle textbasierten PDF-Formularfelder in der zu prüfenden PDF-Datei ob die Länge des Inhaltes des Feldes kleiner oder gleich der im technischen Handbuch angegebenen maximal zulässigen Anzahl an Zeichen ist. Bei PDF-Feldern mit variabler Schriftgröße wird vom Prüfmodul zunächst die verwendete Schriftgröße und anschließend die Anzahl der Zeichen ermittelt. Ist der Inhalt länger als laut technischem Handbuch erlaubt, wird eine Fehlermeldung ausgegeben.

### 5.1.2 Einhaltung der vorgegebenen Schriftgrößen je Textfeld

Das Prüfmodul prüft alle textbasierten PDF - Formularfelder ob die jeweils im technischen Handbuch angegebene Schriftgröße eingehalten wurde. Bei PDF-Feldern mit variabler Schriftgröße bestimmt das Prüfmodul die verwendete Schriftgröße und die Verwendete Anzahl an Zeichen und prüft gegen die im technischen Handbuch entsprechend festgelegte maximale Anzahl an Zeichen. Stimmt die Schriftgröße in einem Feld nicht, wird ein entsprechender Hinweis ausgegeben.

### 5.1.3 Einhaltung des Formats bei Datumsfeldern

Das Prüfmodul prüft die Datumsfelder des PDF's, ob der Inhalt den Formatvorgaben des technischen Handbuchs entspricht. Das Format wird nur geprüft, wenn das entsprechende Datumsfeld nicht leer ist. Es wird ein Fehler ausgegeben, wenn ein Format nicht stimmt.

### 5.1.4 Für alle Textfelder ist die Option „Textlauf“ deaktiviert

Das Prüfmodul prüft, ob bei allen textbasierten PDF-Feldern die Option „DoNotScroll“ aktiviert ist. Für jedes fehlerhaft geprüfte Feld wird eine entsprechende Hinweismeldung ausgegeben.

### 5.1.5 Für alle Felder ist die Option „nur lesend“ gesetzt

Das Prüfmodul prüft, ob bei allen Feldern außer bei „0000\_QES“ die Option „nur lesend“ aktiviert ist. Verläuft die Prüfung bei einem Feld fehlerhaft wird ein entsprechender Fehler ausgegeben.

### 5.1.6 Für alle Felder ist die Option „sichtbar“ gesetzt

Das Prüfmodul prüft, ob bei allen Feldern die Option „sichtbar“ aktiviert ist. Die Optionen „Ausgeblendet“, „Sichtbar, aber drucken nicht möglich“ oder „Unsichtbar, aber Drucken möglich“ dürfen nicht gesetzt sein. Bei einem Verstoß, gibt das Prüfmodul einen Fehler zurück.



### **5.1.7 Das Formular wird gegen die PDF/A-3a – Spezifikationen validiert**

Das Prüfmodul validiert die PDF-Datei gegen die PDF/A3a – Spezifikation. Bei Nichtvalidität gibt das Prüfmodul einen Hinweis aus.

### **5.1.8 Es wird geprüft, ob alle Felder vorhanden und korrekt benannt sind**

Das Prüfmodul prüft, ob alle Felder entsprechend den Vorgaben des techn. Handbuchs vorhanden sind und alle Felder korrekt benannt sind. Bei einem Verstoß gibt das Prüfmodul einen Fehler aus.

## 5.2 Prüfungen im Prüfmodus „Abgleich zwischen digitalem Muster und LDT“

Bei dem Abgleich zwischen PDF und LDT handelt es sich um einen Service, den die KBV zur Verfügung stellt um dem Hersteller die Möglichkeit zu geben die entsprechenden Anforderungen im Anforderungskatalog adäquat umzusetzen und eine Kontrollinstanz im laufenden Betrieb zu haben. Es besteht keine Pflicht diesen Prüfmodus einzusetzen.

Wurde der Modus „Abgleich zwischen digitalem Muster und LDT“ ausgewählt, so wird ein Abgleich zwischen den Inhalten des PDF's und der dazugehörigen LDT-Datei durchgeführt.

Das Prüfmodul kann nur LDT-Dateien verarbeiten, bei denen Auftragsobjekte nicht LDT-Dateiübergreifend sind, pro LDT-Datei nur ein Auftragsobjekt (Satzart 8215) enthalten und sich folgende Objekte in diesem Auftragsobjekt befinden:

- genau ein Feld Veranlassungsgrund (FK 8127)
- mehrere Felder Material (FK 8137)
- genau ein Feld Abrechnungsinformation (FK 8101)
- mehrere Felder Untersuchungsanforderung (FK 8159)

Beim PDF-LDT-Abgleich werden lediglich Hinweise, keine Fehler ausgegeben.

Das Prüfmodul identifiziert das Muster anhand der Metadaten. Und führt nachfolgende Prüffälle aus.

### 5.2.1 Übergreifend

Die folgenden übergreifenden Prüffälle gelten für Muster 10 und Muster 10A gleichermaßen.

#### 5.2.1.1 Kostenträgername stimmt in beiden Dateien überein

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „4134\_Kostenträgername“ mit dem Inhalt des LDT-Felds mit Feldkennung „4134“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### 5.2.1.2 Der Wert für „WOP“ ist in beiden Dateien identisch

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „3116\_WOP“ mit dem Inhalt des LDT-Felds mit Feldkennung „3116“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

### 5.2.1.3 Nachname ist in beiden Dateien identisch

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „3101\_Name“ mit dem Inhalt des LDT-Felds mit Feldkennung „3101“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

### 5.2.1.4 Abgleich für Vorname, Titel, Namenszusatz und Vorsatzwort

Das PDF-Feld „0000\_Titel\_Vorname\_Namenszusatz\_Vorsatzwort“ wird entsprechend der Vorgaben im technischen Handbuch aufgeteilt und geprüft ob

- „Titel“ aus dem PDF-Feld entspricht LDT-Feld mit Feldkennung „3104“
- „Vorname“ aus dem PDF-Feld entspricht LDT-Feld mit FK „3102“
- „Namenszusatz“ aus dem PDF-Feld entspricht LDT-Feld mit FK „3100“
- „Vorsatzwort“ aus dem PDF-Feld entspricht LDT-Feld mit FK „3120“

Ist das nicht der Fall, wird ein entsprechender Hinweistext ausgegeben.

### 5.2.1.5 Der maschinenlesbare Teil des Geburtsdatums ist identisch mit dem Inhalt des LDT Felds 3103

Das Prüfmodul prüft, ob der maschinenlesbare Teil des Inhalts des PDF-Formularfelds „3103\_Geburtsdatum“ mit dem Inhalt des LDT-Felds mit Feldkennung „3103“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

### 5.2.1.6 Abgleich von Straße und Hausnummer

Das PDF-Feld „0000\_Straße\_Hausnummer“ wird entsprechend der Vorgaben im technischen Handbuch aufgeteilt und prüft ob,

- Bei Postfachadresse: PDF-Feld „Postfach“ = LDT FK 3123
- Bei Straßenadresse: Straße(PDF) = LDT FK 3107 und Hausnummer(PDF) = LDT FK 3109

Ist das nicht der Fall, wird ein entsprechender Hinweistext ausgegeben.

### 5.2.1.7 Abgleich PLZ und Ort

Das PDF-Feld „0000\_Wohnsitzlaendercode\_PLZ\_Ort“ wird entsprechend der Vorgaben im technischen Handbuch aufgeteilt und geprüft ob

- Bei Postfachadresse:
  - a) Wohnsitzlaendercode(PDF) = LDT FK 3124 und
  - b) PLZ(PDF) = LDT FK 3112 und
  - c) Ort(PDF) = LDT FK 3122
- Bei Straßenadresse:
  - a) Wohnsitzlaendercode(PDF) = LDT FK 3114 und

- b) PLZ(PDF) = LDT FK 3112 und
- c) Ort(PDF) = LDT FK 3113

Ist das nicht der Fall, wird ein entsprechender Hinweistext ausgegeben.

#### **5.2.1.8 Der maschinenlesbare Teil des Felds „VersicherungsschutzEnde“ ist identisch mit dem Inhalt des LDT-Felds FK 4110**

Das Prüfmodul prüft, ob der maschinenlesbare Teil des Inhaltes des PDF-Formularfelds „4110\_VersicherungsschutzEnde“ mit dem Inhalt des LDT-Felds mit Feldkennung „4110“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.9 Kostenträgerkennung ist in beiden Dateien identisch**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „4111\_Kostentraegerkennung“ mit dem Inhalt des LDT-Felds mit Feldkennung „4111“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.10 VersichertenID ist in beiden Dateien identisch**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „3119\_VersichertenID“ mit dem Inhalt des LDT-Felds mit Feldkennung „3119“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.11 Versichertenart ist in beiden Dateien identisch**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „3108\_Versichertenart“ mit dem Inhalt des LDT-Felds mit Feldkennung „3108“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.12 Besondere Personengruppe ist in beiden Dateien identisch**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „4131\_BesonderePersonengruppe“ mit dem Inhalt des LDT-Felds mit Feldkennung „4131“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.13 DMP-Kennzeichnung ist in beiden Dateien identisch**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „4132\_DMP\_Kennzeichnung“ mit dem Inhalt des LDT-Felds mit Feldkennung „4132“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.14 Abgleich des ASV und TSS-Kennzeichens**

Ist der Inhalt des PDF-Feldes „0000\_weitere\_Kennzeichen“ = 1, dann muss das LDT Feld mit Feldkennung 7303 = 8 sein. Die Prüfung wird für alle vorhandenen Untersu-

chungsanforderungen durchgeführt und gilt als bestanden, wenn alle Anforderungen positiv getestet wurden.

Wenn die Prüfung nicht erfolgreich war wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.15 Betriebsstättennummer muss mit LDT-Feld FK 0222 oder FK 0201 identisch sein**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „0000\_Betriebsstaettennummer“ in den LDT-Feldern mit Feldkennung „0222“ oder „0201“ vorhanden ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.16 Lebenslange Arztnummer muss mit LDT FK 0212 oder FK 0223 identisch sein**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „0212\_LebenslangeArztnummer“ mit dem Inhalt des LDT-Felds mit Feldkennung „0212“ oder „0223“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.17 Ausstellungsdatum muss in beiden Dateien übereinstimmen**

Das Prüfmodul prüft, ob der maschinenlesbare Teil des Inhalts des PDF-Formularfelds „4102\_Ausstellungsdatum“ mit dem Inhalt des LDT-Felds mit Feldkennung „7278“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.18 KBV-Prüfnummer muss in beiden Dateien übereinstimmen**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „0000\_KBV-Prüfnummer“ mit dem Inhalt des LDT-Felds mit Feldkennung „0105“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

**Hinweis:** Die FK0105 findet sich im Objekt sendendes\_system (8151) in den Kopfdaten (8132) in der Satzart 8230 (P-Datenpaket-Header).

#### **5.2.1.19 Abgleich Unfall**

Das Prüfmodul prüft, ob das PDF-Formularfeld „4202\_Unfall“ angekreuzt ist. Wenn das der Fall ist muss das LDT-Feld mit Feldkennung „4202“ = 1 sein. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.20 Laborauftragsnummer muss in beiden Dateien übereinstimmen**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „8311\_Labor\_Auftragsnummer“ mit dem Inhalt des LDT-Felds mit Feldkennung „8311“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### 5.2.1.21 Ausnahmeindikation muss in beiden Dateien identisch sein

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „4229\_Ausnahmeindikation“ in dem LDT-Feld mit Feldkennung „4229“ vorhanden ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### 5.2.1.22 Das Geschlecht muss in beiden Dateien übereinstimmen

Das Prüfmodul prüft, ob

- a) Das PDF-Feld „3110\_Geschlecht\_W“ angekreuzt ist. In dem Fall muss das LDT-Feld mit der Feldkennung 3110 den Wert „W“ haben.
- b) Das PDF-Feld „3110\_Geschlecht\_M“ angekreuzt ist. In dem Fall muss das LDT-Feld mit der Feldkennung 3110 den Wert „M“ haben.
- c) Das PDF-Feld „3110\_Geschlecht\_W“ und „3110\_Geschlecht\_M“ angekreuzt sind. In dem Fall muss das LDT-Feld mit der Feldkennung 3110 den Wert „X“ haben.
- d) In allen anderen Fällen muss der Wert des LDT-Felds gleich „U“ sein.

#### 5.2.1.23 Diagnose muss in beiden Dateien identisch sein.

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „4207\_Diagnose/ Verdachtsdiagnose“ im LDT-Feld mit Feldkennung „4207“ vorhanden ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

Da das Feld „Diagnose“ im LDT mehrfach vorkommen kann, wird dieses Feld so geprüft, dass zunächst das erste Feld FK 8200 ausgelesen wird und geprüft wird, ob der Inhalt des LDT-Feldes in der Zeichenkette im PDF-Feld „4207\_Diagnose/ Verdachtsdiagnose“ vorkommt. Kommt das LDT-Feld im PDF-Feld vor, wird im PDF-Feld der gefundene Text temporär eliminiert. Diese Prozedur wird wiederholt bis alle Textteile eliminiert sind oder keine Felder im LDT mehr übrig sind. Die Prüfung wurde bestanden wenn der Text im PDF vollständig eliminiert werden konnte.

Dabei wird davon ausgegangen, dass Wörter im digitalen Muster immer am Zeilenende enden. Wörter über mehrere Zeilen werden somit nicht unterstützt.

#### 5.2.1.24 Abgleich Abnahmedatum

Das Prüfmodul prüft, ob das Abnahmedatum im maschinenlesbaren Teil des Inhalts des PDF-Feldes „8129\_Abnahmedatum“ mit dem LDT-Feld mit Feldkennung „7278“ identisch ist. Die Prüfung wird für jedes Vorkommen von „Material“ wiederholt und gilt als bestanden wenn für jedes Objekt positiv getestet wurde.

Ist das nicht der Fall, wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.25 Die Abnahmezeit muss in beiden Dateien identisch sein.**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „8219\_Abnahmezeit“ mit den Stunden- und Minutenangaben des LDT-Felds mit Feldkennung „7279“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

Die Prüfung wird für jedes Vorkommen von „Material“ wiederholt und gilt als bestanden wenn für jedes Objekt positiv getestet wurde.

#### **5.2.1.26 Ist „Eilt“ angekreuzt, muss LDT FK 8501=2 sein**

Das Prüfmodul prüft, dass wenn das PDF-Formularfeld „8501\_eilt“ angekreuzt ist, das LDT-feld mit der Feldkennung „8501“ den Wert 2 besitzt. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.1.27 Abgleich Auftragsnummer des Einsenders**

Metadatum „Auftragsnummer des Einsenders“ muss mit LDTFK8310 identisch sein.

Das Prüfmodul prüft, ob der Inhalt des PDF-Metadatum „Auftragsnummer des Einsenders“ mit dem Inhalt des LDT-Felds mit Feldkennung „8310“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

### **5.2.2 Prüffälle für Muster 10**

Die folgenden übergreifenden Prüffälle gelten für Muster 10.

#### **5.2.2.1 Abgleich Kontrolluntersuchung**

Das Prüfmodul prüft, dass wenn das PDF-Formularfeld „4231\_Kontrolluntersuchung\_bekannte\_Infektion“ angekreuzt ist, das LDT-Feld mit der Feldkennung „4231“ den Wert 1 besitzt. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.2.2 Abgleich „Eingeschränkter Leistungsanspruch“**

Das Prüfmodul prüft, dass wenn das PDF-Formularfeld „4204\_eingeschränkter\_Leistungsanspruch“ angekreuzt ist, das LDT-Feld mit der Feldkennung „4204“ den Wert 1 besitzt. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### **5.2.2.3 Abgleich „Erstveranlasser BSNR“**

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „0000\_Erstveranlasser\_BSNR“ in einem der LDT-Felder mit Feldkennung „4217“ oder „4225“ vorhanden ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### 5.2.2.4 Abgleich „Erstveranlasser LANR“

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „0000\_Erstveranlasser\_LANR“ in einem der LDT-Felder mit Feldkennung „4241“ oder „4248“ vorhanden ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### 5.2.2.5 Abgleich „Telefon“

Das Prüfmodul prüft, dass wenn das PDF-Formularfeld „8501\_eilt“ angekreuzt ist, der Inhalt des PDF-Formularfelds „8118\_Telefonnummer“ mit dem Inhalt der LDT-Felder mit den Feldkennungen „7330“ oder „7331“ identisch ist.

Ist das nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### 5.2.2.6 Abgleich „Fax“

Das Prüfmodul prüft, dass wenn das PDF-Formularfeld „8501\_eilt“ angekreuzt ist, der Inhalt des PDF-Formularfelds „8118\_Faxnummer“ mit dem Inhalt des LDT-Feldes mit der Feldkennung „7333“ identisch ist.

Ist das nicht der Fall wird ein entsprechender Hinweis ausgegeben.

#### 5.2.2.7 Abgleich „Befund/Medikation“

Das Prüfmodul prüft, ob der Inhalt des PDF-Formularfelds „4208\_Befund\_Medikation“ mit dem Inhalt des LDT-Feldes mit Feldkennung „4208“ identisch ist. Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben.

Da das Feld „Befund/Medikation“ im LDT mehrfach vorkommen kann, wird dieses Feld so geprüft, dass zunächst das erste Feld FK 4208 ausgelesen wird und geprüft wird, ob der Inhalt des LDT-Feldes in der Zeichenkette im PDF-Feld „4208\_Befund\_Medikation“ vorkommt. Kommt der Inhalt des LDT-Feldes im PDF-Feld vor, wird im PDF-Feld der gefundene Text temporär eliminiert. Diese Prozedur wird wiederholt bis alle Textteile eliminiert sind oder keine Felder im LDT mehr übrig sind. Die Prüfung wurde bestanden wenn der Text im PDF vollständig eliminiert werden konnte.

Dabei wird davon ausgegangen, dass Wörter im digitalen Muster immer am Zeilenende enden. Wörter über mehrere Zeilen werden somit nicht unterstützt.

#### 5.2.2.8 Abgleich „Auftrag“

Es wird davon ausgegangen, dass

- ein oder mehrere Untersuchungsanforderungen im LDT-Auftrag vorhanden sind,
- Pro Untersuchungsanforderung können die Auftragsinformationen den Feldkennungen 7365 & 7366 oder 8412 & 8411 oder 8434 enthalten sein,
- Ist eine der genannten Kombinationen im Objekt „Untersuchungsanforderung“ vorhanden, können die jeweils anderen Kombinationen in diesem Objekt nicht mehr vorkommen.



Da das Feld „Auftrag“ im LDT mehrfach und in verschiedenen Konstellationen (siehe oben) vorkommen kann, um den Inhalt wiederzugeben, werden diese Felder so geprüft, dass zunächst die LDT-Felder der ersten Untersuchungsanforderung ausgelesen werden und geprüft wird, ob der Inhalt der LDT-Felder in der Zeichenkette im PDF-Feld „4205\_Auftrag“ vorkommen. Kommt der Inhalt der LDT-Felder im PDF vor, wird im PDF-Feld der gefundene Text temporär eliminiert. Diese Prozedur wird wiederholt bis alle Textteile eliminiert sind oder keine Untersuchungsanforderungen im LDT mehr übrig sind. Die Prüfung wurde bestanden wenn der Text im PDF vollständig eliminiert werden konnte.

Dabei wird davon ausgegangen, dass Wörter im digitalen Muster immer am Zeilenende enden. Wörter über mehrere Zeilen werden somit nicht unterstützt.

Ist dies nicht der Fall wird ein entsprechender Hinweis ausgegeben

#### 5.2.2.9 Abgleich „Leistungsart“

Es muss gelten:

- a) Wenn PDF-Feld „4221\_Kurativ“ angekreuzt, dann LDT-Feld 4221 = 1
  - b) Wenn PDF-Feld „4221\_Praeventiv“ angekreuzt, dann LDT-Feld 4221 = 2
  - c) Wenn PDF-Feld „4221\_belegaerztliche\_Behandlung“ angekreuzt, dann LDT-Feld 4221 = 4
  - d) Wenn PDF-Feld „4221\_ESS“ angekreuzt, dann LDT-Feld 4221 = 3
- Ist das nicht der Fall, wird ein entsprechender Hinweis ausgegeben.

#### 5.2.3 Prüffälle für Muster 10A

Die folgenden Prüffälle gelten für den Abgleich Muster 10A und LDT.

##### 5.2.3.1 Abgleich „Auftrag“

Das Prüfmodul prüft, ob alle auf dem PDF angekreuzten Felder „4205\_Auftrag<ID>“ in der LDT-Datei unter der Feldkennung „8410“ in einem Untersuchungsanforderungsobjekt hinterlegt sind.

Wurde das Feld „4205\_Auftrag61“ angekreuzt, so prüft das Prüfmodul, ob der Inhalt des PDF-Feldes „4205\_Auftrag61\_sonstige Auftraege“ Bestandteil des LDT-Feldes mit der Feldkennung 8434 in einem Untersuchungsanforderungsobjekt ist.

**Beispiel:** Im PDF sind die Felder „4205\_Auftrag2“, „4205\_Auftrag6“ und „4205\_Auftrag35“ angekreuzt. Im LDT wird geprüft, ob folgende Felder vorhanden sind: „8410=2“, „8410=6“ sowie „8410=35“.

### 5.2.3.2 Abgleich „Leistungsart“

Es wird geprüft:

- a) Wenn PDF-Feld „4221\_Kurativ“ angekreuzt, dann muss LDT-Feld 4221 = 1
- b) Wenn PDF-Feld „4221\_Praeventiv“ angekreuzt, dann muss LDT-Feld 4221 = 2
- c) Wenn PDF-Feld „4221\_belegaerztliche\_Behandlung“ angekreuzt, dann muss LDT-Feld 4221 = 4

Ist das nicht der Fall, wird ein entsprechender Hinweis ausgegeben.

## 5.3 Prüfungen im Prüfmodus „LDT“ bzw. „LDT Verzeichnisprüfung“

Es werden die Vorgaben in der LDT 3.0 – Datensatzbeschreibung geprüft. Hierbei wird über die Konfiguration gesteuert, welcher Regelsatz (basis, kbv, qms, gesamt) verwendet wird.