



Kassenärztliche  
Bundesvereinigung

Körperschaft des öffentlichen Rechts

## ***IT in der Arztpraxis***

### ***Handbuch KBV-Prüfmodul XPM***

[KBV\_ITA\_AHEX\_Handbuch\_Pruefmodul]

Dezernat Digitalisierung und IT

10623 Berlin, Herbert-Lewin-Platz 2

Kassenärztliche Bundesvereinigung

Version 1.34  
Datum: 15.08.2023  
Kennzeichnung: Öffentlich  
Status: In Kraft

## DOKUMENTENHISTORIE

Version	Datum	Autor	Änderung	Begründung	Seite
1.34	15.08.2023		Java-Version erhöht		7
1.33	10.02.2022		Kapitel Zeichensatzkonverter überarbeitet Kapital Anhang überarbeitet	Klassennamen haben sich geändert	<u>22</u> 24
1.32	10.01.2022		Minimale Java Version geändert in 11, Abbildungen aktualisiert	Kompilierung	20
1.31	03.02.2016		Minimale Java Version geändert in 1.8	Kompilierung	
1.30	30.04.2015		Abschnitt 3.1.4.1 entfallen	Schalter zeichensatz_pruefung ist obsolet	
1.29	19.07.2012		Dokument an Standard Struktur angepasst		
1.28	19.01.2012		Neuer Konfigurationsparameter für Protokolldatei.	Programmerweiterung	12
1.27	27.10.2011		Java Version 1.5 wird nicht mehr unterstützt. Neue Protokolldatei.	Programmerweiterung	12

## INHALTSVERZEICHNIS

<b>DOKUMENTENHISTORIE</b>	<b>2</b>
<b>INHALTSVERZEICHNIS</b>	<b>3</b>
<b>ABBILDUNGSVERZEICHNIS</b>	<b>5</b>
<b>TABELLENVERZEICHNIS</b>	<b>6</b>
<b>1 EINLEITUNG</b>	<b>7</b>
1.1 Begriffsklärung .....	7
1.2 Systemvoraussetzungen .....	8
<b>2 VERZEICHNISSTRUKTUR</b>	<b>9</b>
2.1 Ordner 'Bin' .....	9
2.2 Ordner 'Daten' .....	9
2.3 Ordner 'Doku' .....	9
2.4 Ordner 'Geprueft/Abgelehnt' .....	9
2.5 Ordner 'Geprueft/NichtAbgelehnt' .....	9
2.6 Ordner 'Kbvtab' .....	9
2.7 Ordner 'Konfig' .....	10
2.8 Ordner 'Listen' .....	10
2.9 Ordner 'Schema' .....	10
<b>3 ARBEITEN MIT XPM</b>	<b>11</b>
3.1 Technische Hinweise .....	11
3.1.1 Abschnitt: Allgemeiner Teil .....	11
3.1.2 Abschnitt: Eingabedateien .....	14
3.1.3 Abschnitt: Ausgabedateien .....	14
3.1.4 Abschnitt: Schalter .....	16
3.1.5 Umgebungsvariablen .....	16
3.2 Arbeiten im Kommandozeilenmodus .....	18
3.2.1 Übergabeparameter .....	18
3.2.2 Beispiele .....	19

<b>3.3</b>	<b>Arbeiten mit grafischer Oberfläche .....</b>	<b>19</b>
3.3.1	Prüfdatei selektieren .....	20
3.3.2	Prüfung starten .....	21
3.3.3	Prüfung abbrechen .....	21
<b>4</b>	<b><u>ZUSATZPROGRAMME</u> .....</b>	<b>22</b>
4.1	Zeichensatzkonverter .....	22
<b>5</b>	<b><u>ANHANG</u> .....</b>	<b>24</b>
5.1	Starten eines Java-Programms .....	24
5.2	Starten des Prüfmoduls aus einem Java-Programm.....	24
<b>6</b>	<b><u>ANHANG: KBV-PRÜFMODUL XPM FÜR DMP</u> .....</b>	<b>25</b>
6.1	Konfiguration der XPM-Meldungen .....	25
6.2	Meldungsliste .....	25
6.3	Hinweise .....	26

## ABBILDUNGSVERZEICHNIS

Abbildung 1: XPM Prüfmodul für KVDT Prüfung.....	20
Abbildung 2: XPM Prüfmodul für Brustkrebs Prüfung .....	20

## TABELLENVERZEICHNIS

Tabelle 1: ErrorLevel .....	15
Tabelle 2: Übergabeparameter .....	18
Tabelle 3: Übergabeparameter .....	23

# 1 Einleitung

Dieses Dokument gibt einen kurzen Überblick über die Arbeitsweise und eine genaue Beschreibung zur Installation und Ausführung des neuen KBV-Prüfmoduls XPM.

XPM ist ein Prüfprogramm für Datensätze, die einer definierten Schnittstellenbeschreibung entsprechen müssen. Als Ergebnis der Prüfung wird unter anderem ein Prüfprotokoll erstellt welches Informationen über den Zustand der Prüfdatei enthält. Das Ergebnis der Prüfung ist je nach Zustand des Prüflings folgender Status: Ok, Warnung, Fehlerhaft oder Abbruch möglich.

XPM ist auf allen Computersystemen lauffähig, für die die Java Laufzeitumgebung in der Version 17 verfügbar ist.

## 1.1 Begriffsklärung

### Was ist ein XPM-Paket?

Ein XPM-Paket ist ein schnittstellenspezifisches Prüfmodulpaket, das aus folgenden Teilen besteht:

- XPM-Kernel (JAVA-Programm zum Interpretieren eines XPM-Prüfprojektes)
- XPM-Prüfprojekt (schnittstellenspezifische Konfiguration)
- ReleaseNotes (Textdatei, die alle Informationen zur Version des Prüfmodulpakets enthält)
- GUI (Aufruf des Prüfmoduls als graphische Benutzungsoberfläche)
- Batchdateien und Shellskripte (Beispiele für den commandline-orientierten Aufruf in verschiedenen Modi)

### Begriffserläuterungen:

- **XPM:** Das X steht für beliebige Daten- bzw. Schnittstellenarten, PM steht für Prüfmodul, das XPM steht als Synonym für ein (durch ein Prüfprojekt) konfigurierbares Prüfmodul.
- **XPM-Paket:** Schnittstellenspezifisches Prüfmodulpaket.
- **XPM-Paket\_xxx\_20aa.b.c:** Schnittstellenspezifisches Prüfmodulpaket der Schnittstelle xxx der XPM-Paket-Version c für das Quartal b des Jahres aa, wobei es sich um die Auslieferung Nr. c für das Quartal b handelt, auch "das aktuelle xxx-Prüfmodul". Definiert die Schnittstelle eine erstmalige Dokumentation und Verlaufsdokumentation bzw. Erstdokumentation und Folgedokumentation so enthält das Prüfpaket Prüfungen für beide Schnittstellen.
- **XPM-Prüfprojekt:** Tabellen und Codeelemente, die die schnittstellenspezifische Konfiguration enthalten, schnittstellenspezifischer Bestandteil des XPM-Pakets
- **XPM-Kernel:** Programm zum Interpretieren eines XPM-Prüfprojektes, der Kernel ist ohne schnittstellenspezifisches Prüfprojekt, welches die schnittstellenspezifische Konfiguration enthält, nicht verwendbar; XPM-allgemeiner Bestandteil des XPM-Pakets
- **KVDT-XPM-Paket:** Das KVDT-XPM-Paket löste das alte DOS basierte Prüfmodul ab (auch ADT-Prüfmodul genannt, da ADT der wesentliche Bestandteil des KVDT ist). Die offiziell richtige Bezeichnung ist das "KBV-Prüfmodul".

- **"Prüfmodul-Version"**: Ein XPM-Paket, eine Prüfmodul-Version ist immer Schnittstellen-spezifisch.
- **"das neue KBV-Prüfmodul"**: Hiermit ist meist die neueste Version des KVDT-XPM-Pakets gemeint.
- **"Java-Prüfmodul"**: Früherer Arbeitsname für den XPM-Kernel, weil dafür die Programmiersprache JAVA verwendet wird
- **"Gesamtpaket xxx Vf.gg"**: Das Gesamtpaket enthält alle relevanten Dokumente und Software für eine Schnittstelle bzw. Indikation. Achtung! Die Versionsnummer eines Gesamtpakets ist nicht zu verwechseln mit der Versionsnummer der Schnittstelle oder des Prüfmoduls

## 1.2 Systemvoraussetzungen

XPM ist eine Applikation für ein 32 oder 64bit-System.

Benötigt wird ein Computersystem, das leistungsmäßig mit einem IBM-kompatiblen PC, 128 MB Hauptspeicher und einem Pentium-Prozessor 500 oder höher vergleichbar ist.

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



## 2 Verzeichnisstruktur

Im Installationsverzeichnis befinden sich Batchdateien bzw. Shellskripte, die das KBV-Prüfmodul mit unterschiedlichen Optionen bzw. Konfigurationsdateien ausführen.

Die Verzeichnisstruktur des KBV-Prüfmoduls hat folgenden Aufbau:

### 2.1 Ordner 'Bin'

Dieser Ordner beinhaltet alle Java-Archive, die zur Ausführung des XPM-Pakets benötigt werden.

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

### 2.2 Ordner 'Daten'

Dieser Ordner enthält Prüfdateien

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

### 2.3 Ordner 'Doku'

Dieser Ordner enthält die Dokumentation zum XPM

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

### 2.4 Ordner 'Geprueft/Abgelehnt'

Dieser Ordner dient als Ablage für **nicht** korrekt geprüfte Dateien. Dieser Ordner kann bei einigen Prüfprojekten entfallen.

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

### 2.5 Ordner 'Geprueft/NichtAbgelehnt'

Dieser Ordner dient als Ablage für korrekt geprüfte Dateien. Dieser Ordner kann bei einigen Prüfprojekten entfallen.

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

### 2.6 Ordner 'Kbvtab'

Dieser Ordner enthält alle binären Steuerdateien und Stammdaten.

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

## 2.7 Ordner 'Konfig'

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

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

## 2.8 Ordner 'Listen'

In diesem Ordner werden alle Ausgaben des Prüfmoduls generiert.

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

## 2.9 Ordner 'Schema'

In diesem Ordner befinden sich die XML-Schemadateien, die das Prüfformat definieren. Die XML-Schemadateien (\*.xsd) dürfen **nicht** verändert werden. In einigen Prüfumgebungen können sich die XML-Schemas zusätzlich in dem Ordner 'Konfig' befinden.

In einigen XPM-Prüfpaketen kann dieser Ordner unter einem anderen Namen konfiguriert worden sein.

## 3 Arbeiten mit XPM

### 3.1 Technische Hinweise

XPM steht als einheitliches Werkzeug zur Prüfung möglichst vieler Eingangs- und Ausgangsdaten zur Verfügung. Dabei verarbeitet es unterschiedliche Dateiformate wie xDT und XML. Weitere Eingangsformate werden folgen. XPM läuft in einer Java Laufzeitumgebung und kann somit auf diversen Betriebssystemen eingesetzt werden.

XPM besteht im Kern aus einem einzigen "universellen" Steuermodul (pruefmodul.jar) sowie einem "speziellen" Prüfmodul (pruefungen.jar).

Zur Prüfung eines speziellen Eingangsformates benötigt das XPM Informationen über den Dokumentenaufbau. Diese Informationen erhält XPM auch zur Laufzeit durch Auswertung (Interpretation) eines XML-Schemas.

Zu einem Prüflauf können weitere Dateien wie z. B. Schlüsseltabellen und Stammdateien hinzugezogen werden. Diese Optionen sind über eine Konfigurationsdatei einstellbar.

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

Die Konfiguration eines Prüfmodullaufs erfolgt mit Hilfe einer XML-Konfigurationsdatei. Hier folgt eine allgemeine Einführung in den Aufbau einer Konfigurationsdatei. In den jeweiligen Konfigurationsdateien selbst existiert zu jedem Konfigurations-Element 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 Konfigurationsdatei auf verschiedenen Betriebssystemen (Windows, Unix, Linux, ...). Relative Pfadangaben werden als relativ zum Installationsverzeichnis betrachtet. Die Konfigurationsdatei wird in 4 Abschnitte eingeteilt.

#### 3.1.1 Abschnitt: Allgemeiner Teil

Im allgemeinen Teil werden allgemeine Informationen zum Prüflauf eingestellt.

##### 3.1.1.1 Installationsverzeichnis

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

Z.B.: `<pruefpfad>C:/KBV_Pruefmodul/</pruefpfad>`

##### 3.1.1.2 Prüfdatenverzeichnis

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

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

### 3.1.1.3 Geprüft-OK-Verzeichnis

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

Z.B.: `<okdaten>Fehlerfrei/</okdaten>`

### 3.1.1.4 Ausschussverzeichnis

Bei eingeschalteter Option `-m` wird die geprüfte Datei mit Status 'abgelehnt' bzw. 'abbruch' in dieses Verzeichnis verschoben. Pfadangabe erfolgt im Element „**fehlerdaten**“.

Z.B.: `<fehlerdaten>Fehler/</fehlerdaten>`

### 3.1.1.5 Temporäres Verzeichnis

Während der Prüfung (XPM-KVDT-Paket) werden einige Daten in temporären Dateien gehalten. Dieses Verzeichnis wird durch Pfadangabe im Element „**tempdaten**“ spezifiziert. Fehlt dieser Schalter so wird das standardmäßige, betriebssystem-spezifische Verzeichnis genommen.

Z.B.: `<tempdaten>Fehler/</tempdaten>`

### 3.1.1.6 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 zu genauen Fehlerursache.

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

### 3.1.1.7 Dateifilter

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

Z.B.: `<datei_filter>*.ED2</datei_filter>`

### 3.1.1.8 Datensatzbeschreibung

Die Datensatzbeschreibung wird in Form einer XML-Schemadatei spezifiziert.

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

Z.B.: `<pruefschema>Konfig/koloskopie.xsd</pruefschema>`

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

### 3.1.1.9 Info

Diese Einstellung ermöglicht es das Protokollieren der Informationsmeldungen ein- bzw. auszuschalten. Die Konfiguration wird im Element „info“ festgelegt. Diese Einstellung ist unabhängig von der Konfiguration zur Protokollierung von Warnungen s.u.

Wertebereich: „ja“, „nein“.

Z.B. `<info>ja</info>`

### 3.1.1.10 Warnungen

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

Wertebereich: „ja“, „nein“.

Z.B. `<warnungen>ja</warnungen>`

### 3.1.1.11 Fehler

Diese Einstellung ermöglicht es das Protokollieren der Fehler ein- bzw. auszuschalten. Die Konfiguration wird im Element „fehler“ festgelegt.

Wertebereich: „ja“, „nein“.

Z.B. `<fehler>ja</fehler>`

**Warnung:** Eine Deaktivierung der Fehlerausgabe sollte nur zu Testzwecken genutzt werden. Ansonsten kann es vorkommen, dass das Fehlerprotokoll den Status ‚abgelehnt‘ hat obwohl keine Fehlermeldungen protokolliert wurden.

### 3.1.1.12 Begrenzung der Fehlermeldungen

Standardmäßig begrenzt XPM 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>nein</fehler_begrenzen>`

### 3.1.1.13 Abbruch nach Überschreitung einer bestimmten Anzahl von Fehlern

Das XPM kann während der Laufzeit abgebrochen werden, wenn eine vorher festgelegte Anzahl von Fehlern erreicht wird. Diese Einstellung ermöglicht die Prüfung von sehr großen Dateien vorzeitig abubrechen, wenn bereits vor dem Ende der Prüfung feststeht, dass eine Prüfung nicht erfolgreich sein wird.

Z.B. `<fehler_abbruch>10</fehler_abbruch>`

### 3.1.1.14 Spaltenangabe

Eine Einstellung, ob zusätzlich zu den Zeilenangaben auch die Spaltenangaben bei den jeweiligen Meldungen in der Protokolldatei erscheinen sollen, werden im Element „spalten“ festgelegt.

Wertebereich: „ja“, „nein“.

Z.B. `<spalten>nein</spalten>`

### 3.1.1.15 Dokumentation

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

Z.B. `<dokumentation>Doku/E014_XPM_Anwenderhandbuch.pdf</dokumentation>`

## 3.1.2 Abschnitt: Eingabedateien

Im Abschnitt Eingabedateien werden die Pfade zur Steuertabelle und Stammdatei(en) festgelegt. Die Steuertabelle für das XPM, im Element „kbv\_tabelle“ spezifiziert, muss in der Konfigurationsdatei angegeben werden.

Z.B.:

```
<kbv_tabelle>Kbvtab/kbv_tabelle.bin</kbv_tabelle>
```

Die Steuertabelle und die Prüfbibliothek (z.B. die Datei pruefungKVDT\_V1.xx\_Q07x\_x.jar im Ordner Bin) **müssen** aus ein und dem selben XPM-Paket stammen. Andernfalls kommt es zum Programmabbruch mit der Meldung: „Das Laden der Prüfklasse 'de.kbv.pruefmodul.generiert.xyz.XPMAdapter'ist fehlgeschlagen!“

Über das Attribut Einlesen wird das Ladeverhalten des XPM's für die jeweilige Eingabedatei gesteuert. Ist hier der Wert „ja“ eingetragen wird die Stammdatei zur Prüfung herangezogen. Beim Wert „nein“ erfolgt keinerlei Prüfung gegen diese Stammdatei. Der Wert „auto“ veranlasst das XPM selbständig nach einer gültigen Stammdatei zu suchen, diese dann zu laden und für eine Prüfung bereitzustellen. Diese Einstellung wird für Stammdaten benötigt, die in mehreren Versionen vorliegen, wie z.B. die GOS, die für jeden KV-Bereich einzeln vorliegt. Die Einstellung „auto“ darf nur im Zusammenhang mit Stammdaten verwendet werden, die die automatische Handhabung auch vorsieht, andernfalls würde die Prüfung gegen diese Stammdatei ausgeschaltet werden. Wird das Attribut Einlesen ausgelassen, so wird die Stammdatei standardmäßig geladen und zur Prüfung herangezogen.

Z.B.:

```
<KTStamm Einlesen="ja">Kbvtab/KTStamm.bin</KTStamm>
```

**Warnung:** Vor dem Ausschalten der Prüfung gegen eine Stammdatei wird dringend abgeraten, da dies das Prüfergebnis verfälschen kann!

## 3.1.3 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 (Spätere Versionen)
- 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

Z.B.:

```
<FehlerListe Format="PDF">Listen/Fehler.pdf</FehlerListe>
```

PDF Dateien können angezeigt und gedruckt werden. Alles, was dazu benötigt wird, ist der Adobe Reader®, der kostenlos unter <http://www.adobe.de/products/acrobat/readstep2.html> heruntergeladen werden kann. XPM erzeugt PDF Dokumente, die vom Acrobat Reader® ab der Version 5.0 und 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, der den Errorlevel der Prüfung kennzeichnet. Es werden folgende Präfixe verwendet:

Errorlevel	Präfix
0	Ok_
1	Warnung_
2	Fehler_
3	Abbruch_

Tabelle 1: 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 `$_DATEI_NAME` verwendet wird, dann ersetzt XPM diese Variable durch den Namen der Prüfdatei. Fehlt die Variable `$_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.

### 3.1.3.1 Statistikdatei

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

Z.B.:

```
<StatistikListe Format="PDF">Listen/Statistik.pdf</StatistikListe>
```

In der Statistikliste werden 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 '%' enthalten. Dies ist kein Programmfehler sondern nur ein Hinweis darauf, dass diese Meldung variable Inhalte enthält, die erst zur Laufzeit ermittelt werden und unterschiedliche Ausprägungen enthalten kann.

Z.B.:

Zu der GNR '%' wurde keine Angabe im Feld '%' gemacht.

### 3.1.3.2 Zusätzliche Ausgabelisten

In vielen Prüfpaketen existieren zusätzliche Ausgabelisten, die Informationen bzw. Statistiken über den Prüfling enthalten.

Alle diese Listen werden über einen entsprechenden Schalter im Abschnitt Ausgabedateien spezifiziert.

Z.B.:

```
<XXXListe Format="PDF">Listen/ZusatzStatistik.pdf</XXXListe>
```

Das Kürzel XXX steht hier für eine bestimmte Liste des jeweiligen Prüfpaketes.

### 3.1.4 Abschnitt: Schalter

Über die Konfigurationsschalter kann der Programmablauf modifiziert werden. Die Schalter sind Einstellungsmöglichkeiten, die prüfprojekt-spezifisch sind. Bitte beachten Sie auch die Kommentare in den jeweiligen Konfigurationsdateien selbst.

Im KVDT-Projekt wird beispielsweise folgender Schalter definiert:

```
<adt_interne_patientennummer>ja</adt_interne_patientennummer>
```

### 3.1.5 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 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 zugreift.

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=KVDT.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 bekannt gegeben.

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

## 3.2 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.

### 3.2.1 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 Einzelaufmodus gestartet. XPM verarbeitet alle Dateien eines Eingangsverzeichnisses 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 Konfigurationmöglichkeiten 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. Die 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 Beispiel 4 an. Dieser Übergabeparameter ist optional.
-s	Das Prüfmodul wird im s.g. Servermodus gestartet. XPM verarbeitet im 30 Sekunden-Takt Dateien eines Eingangsverzeichnisses. 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.

**Tabelle 2: Übergabeparameter**

Um eine Prüfung durchzuführen muss einer der Schalter -f, -z bzw. -s angegeben werden.

### 3.2.2 Beispiele

Nun folgen 3 Beispiele für den Aufruf des XPM.

#### 1. Prüfen einer einzelnen Datei.

Übergabeparameter:

```
-c Konfig/konfig.xml -f Daten/X0112345.CON
```

XPM liest die Konfigurationsdatei 'Konfig/konfig.xml' ein und prüft die Datei 'Daten/X0112345.CON'. Anschließend wird XPM beendet.

#### 2. Prüfen eines Verzeichnisses.

Übergabeparameter:

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

XPM 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 beendet.

#### 3. Prüfen eines Verzeichnisses (Kein automatisches Beenden).

Übergabeparameter:

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

XPM 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.

#### 4. Setzen des Pfades einer Ausgabeliste und deren Format.

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 XML gesetzt.

## 3.3 Arbeiten mit grafischer Oberfläche

Die grafische Oberfläche stellt eine Alternative zum Kommandozeilen-Programm. Für die KVDT-Prüfung ist die grafische Oberfläche ausschließlich zur Prüfung einzelner Dateien ausgelegt. Für die anderen Schnittstellen-Prüfungen (Brustkrebs, DM1, DM2, VDX und andere) können sowohl Verzeichnisse als auch Zip-Archive geprüft werden.

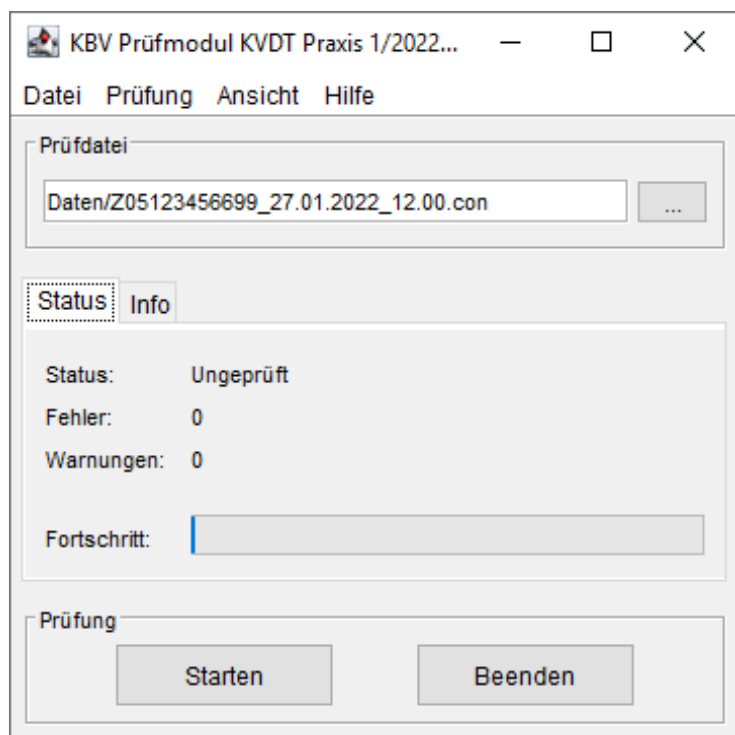


Abbildung 1: XPM Prüfmodul für KVDt Prüfung

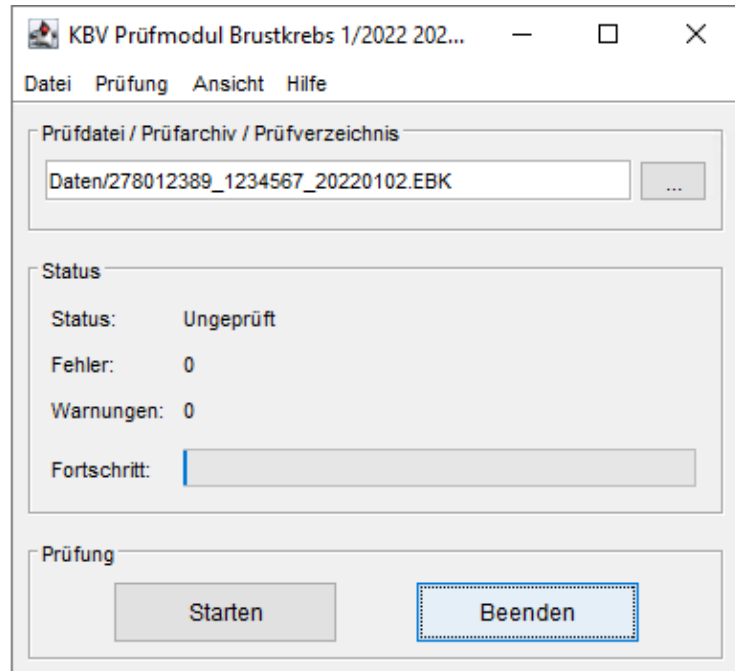


Abbildung 2: XPM Prüfmodul für Brustkrebs Prüfung

### 3.3.1 Prüfdatei selektieren

Bitte drücken Sie den '...' -Knopf und wählen Sie die gewünschte Prüfdatei aus.

### **3.3.2 Prüfung starten**

Bitte drücken Sie den 'Starten'-Knopf. Während der Prüfung zeigt der Fortschrittsbalken den aktuellen Fortschritt. Nach erfolgter Prüfung erscheint eine Meldung mit einem entsprechenden Prüfstatus.

### **3.3.3 Prüfung abbrechen**

Während der Prüfung besteht die Möglichkeit über den 'Abbrechen'-Knopf eine Prüfung vorzeitig abzuberechnen.

## 4 Zusatzprogramme

### 4.1 Zeichensatzkonverter

Der Zeichensatz-Konverter wandelt komplette Abrechnungsdateien in einen gewünschten Zeichensatz um. Für XML-Dateien kann er nicht verwendet werden, da bei diesen im Gegensatz zu KVDT-Dateien nicht die Zeichenkodierung anhand des Dateinamens ermittelt werden kann.

Hier ein Beispiel für einen Aufruf:

```
java -Xmx300m
    -Dfile.encoding=8859_1
    -classpath "Bin/xpm-core-4.2.15.jar"
    de.kbv.xpm.core.converter.CharsetConverter
-q Daten\Z05123456699_31.03.2010_12.00.CON
-z Daten\Umgewandelt\
-s IBM
-l
```

Der Aufruf von Java-Programmen wird im Kapitel 5.1 ausführlich erläutert.

Der Parameter 'de.kbv.xpm.core.converter.CharsetConverter' ist der Name einer Klasse, die den Konverter startet. Der optionale Parameter '-l' bewirkt, dass die Ursprungsdatei (Z05123456699\_31.03.2010\_12.00.CON) nach der Verarbeitung gelöscht wird. Die konvertierte Datei wird also unter den Namen X05123456699\_31.03.2010\_12.00.CON erzeugt.

Wird kein Dateiname für die Ausgabedatei vergeben (Parameter -z) so wird der Name konvertierten Datei wird folgendermaßen gebildet:

Der erste Buchstabe im Dateinamen symbolisiert den Zeichensatz.

- S bei 7Bit
- X bei IBM
- A bei ISO-8859-1
- Z bei ISO-8859-15

Der restliche Dateiname entspricht dem ursprünglichen Dateinamen.

Bei der Abarbeitung kompletter Verzeichnisse werden nur \*.CON bzw. \*.WTK Dateien berücksichtigt.

Übergabeparameter	Beschreibung
-q <Dateiname>	Der Dateiname der zu konvertierenden Datei bzw. Verzeichnisname.
-z <Dateiname>	Der Dateiname der umgewandelten Datei bzw. Verzeichnisname.
-s <Zeichensatz>	Der Zeichensatz in den konvertiert werden soll. Mögliche Werte: 7Bit IBM ISO-8859-1 ISO-8859-15

Übergabeparameter	Beschreibung
-b	Die konvertierte Datei wird unter dem Namen der Ursprungsdatei abgelegt. Dieser Parameter wird nur dann berücksichtigt, wenn die Aktion nicht der Dateinamenskonvention widerspricht. Dateien mit Dateinamen, die mit dem Zeichen A, S oder X beginnen, können nicht unter Ihrem Ursprungsnamen abgelegt werden, da dies der Dateinamenskonvention widerspricht. Dieser Übergabeparameter ist optional.
-h	Ausgabe eines Hilfetextes.
-l	Die Ursprungsdatei wird nach der Verarbeitung gelöscht. Dieser Übergabeparameter ist optional.
-t	Eingangsdateien sind beliebige Textdateien, kein xDT-Format

**Tabelle 3: Übergabeparameter**

## 5 Anhang

### 5.1 Starten eines Java-Programms

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

Hier ein Beispiel für einen Aufruf:

```
java -Xmx300m
    -Dfile.encoding=8859_1
    -classpath "Bin/xpm-dmp-dm2-2022.2.0;..."
    de.kbv.xpm.modul.dmp.dm2.StartGUI
    -c Konfig/konfig.xml -f Daten/X1312345.CON
```

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\_1' stellt den entsprechenden Zeichensatz ein und ermöglicht hier die Verwendung von deutschen Umlauten.

Der Parameter '-classpath "Bin/xpm-dmp-dm2-2022.2.0.jar;..."' stellt alle Java-Archive zusammen, die für den Programmablauf benötigt werden.

Der Parameter 'de.kbv.xpm.modul.dmp.dm2.StartGUI' ist der Name einer Klasse, die das XPM startet.

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

### 5.2 Starten des Prüfmoduls aus einem Java-Programm

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



## 6 Anhang: KBV-Prüfmodul XPM für DMP

Dieses Kapitel gibt einen kurzen Überblick über die Erweiterungen des KBV-Prüfmoduls XPM für DMP-Dokumentationen gegenüber anderen XPM-Prüfpaketen.

### 6.1 Konfiguration der XPM-Meldungen

Dieses Kapitel beschreibt die Vorgehensweise bei der Änderung der Definition der XPM Meldungen und die Einbindung dieser in das XPM-Paket.

XPM ermöglicht es, den KV'en bzw. den Datenannahmestellen eigene Meldungstexte, Error-levels und die Anzahl der maximal auszugebenen Meldungen zu definieren.

### 6.2 Meldungsliste

Die Meldungsliste Meldungen.xml befindet sich im Verzeichnis 'konfig'. Diese Meldungsliste kann mit einem Texteditor bearbeitet werden.

Beispiel: Die Meldung DM1-FEHL wie sie standardmäßig definiert ist.

```
<meldung>                                <!-- Element Meldung -->
  <nummer>DMP-FEHL</nummer>             <!-- Meldungsnummer, darf NICHT verändert werden -->
  <typ>Fehler</typ>                       <!-- Errorlevel -->
  <text>Die Angabe zum Feld '%s' fehlt.</text> <!-- Meldungstext mit Platzhaltern %s -->
  <maxcount>5</maxcount>                 <!-- Anzahl der maximal auszugebenen Meldungen -->
</meldung>
```

Nun soll die Meldung insgesamt 10 mal pro Prüflauf gemeldet werden

```
<meldung>
  <nummer>DMP-FEHL</nummer>
  <typ>Fehler</typ>
  <text>Die Angabe zum Feld '%s' fehlt.</text>
  <maxcount>10</maxcount>               <!-- geänderter Wert -->
</meldung>
```

Nach der Änderung der XML-Meldungsliste muss in der Konfigurationsdatei 'konfigED.xml' bzw. 'konfigVD.xml' der Schalter (Element) meldungsliste das Kommentarzeichen wieder gelöscht werden. XPM aktualisiert die Meldungen zur Laufzeit. Eine vorhergehende Konvertierung ist nicht mehr nötig.

Die Anzahl der maximal auszugebenen Meldungen hat nur dann Auswirkungen auf die Protokollierung, wenn der Konfigurationsschalter 'fehler\_begrenzen' auf den Wert 'ja' eingestellt wird.

## 6.3 Hinweise

- Die Meldungsnummern dürfen NICHT geändert werden.
- Von der Änderung der Errorlevels wird definitiv abgeraten, da dieses das Prüfergebnis beeinflussen kann.
- Wird dem Konfigurationsschalter 'fehler\_begrenzen' der Wert 'nein' zugewiesen, so werden alle aufgetretenen Meldungen protokolliert.
- Wird kein Wert (bzw. der Wert -1) im Element maxcount angegeben, so wird die Meldung so oft ausgegeben wie die Regelverletzung auftritt.