

# DALE-UV PDF-Generator

**gültig ab Version 6.0.0**  
**Stand: 01.04.2022**

**Verfasser: DGUV / DALE-UV**

# Inhaltsverzeichnis

<b>1</b>	<b>Historie.....</b>	<b>3</b>
<b>2</b>	<b>Einleitung .....</b>	<b>4</b>
<b>3</b>	<b>Installation.....</b>	<b>5</b>
3.1	<i>Voraussetzungen.....</i>	5
3.1.1	<i>Hardware-Voraussetzungen.....</i>	5
3.1.2	<i>Software-Voraussetzungen.....</i>	5
3.2	<i>Installation PDF-Generator.....</i>	5
3.2.1	<i>Grundinstallation .....</i>	5
<b>4</b>	<b>Bedienung .....</b>	<b>7</b>
4.1	<i>Einrichten einer neuen Java VM.....</i>	7
4.2	<i>Verwenden einer bestehenden Java VM .....</i>	7
4.3	<i>Java VM Funktionalität prüfen.....</i>	8
4.4	<i>Ausführen des PDF-Generators .....</i>	8
4.5	<i>Aufruf-Syntax .....</i>	10
4.6	<i>Rückgabewerte des Aufrufs .....</i>	11
<b>5</b>	<b>Test des PDF-Generators .....</b>	<b>12</b>
<b>6</b>	<b>Java-Schnittstelle .....</b>	<b>14</b>
6.1	<i>Beschreibung.....</i>	14
6.2	<i>Beispiel-Listing .....</i>	14
<b>7</b>	<b>Log-Ausgabe umkonfigurieren (Log4j).....</b>	<b>15</b>

# 1 Historie

Version	Änderung	Autor	Datum
0.1	Erstellung	Krüger	01.04.07
0.2	Ergänzungen, Anpassungen	Wohlrab	03.05.07
1.0	QS und Freigabe	Geburtig	30.05.07
1.1	Updates und Fehlerkorrektur Stufe 2	Geburtig	28.08.07
1.2	Update für Berichtsversion 08:1:01	Geburtig	03.06.08
1.3	Update für Berichtsversion 13:1:01	Daenicke	31.07.13
4.0	Neuer PDF-Generator für Berichtsversion 17:1:01	Müller	31.07.17
4.1	- Umstellung von .exe-Anwendung auf Verwendung mit eigener Java VM - Fehlerkorrekturen in Berichtsversion / Verkleinerung Dateigröße PDFs	Nöthlichs/Penkaitis	05.09.19
5.0	Neuer PDF-Generator für Berichtsversion 19:4:01	Penkaitis	05.09.19
6.0	Neuer PDF-Generator für Berichtsversion 22:2:01 Hinweis Verwendung Encoding ab OpenJDK 18.	Penkaitis	01.04.22

Tabelle 1: Änderungshistorie

## 2 Einleitung

Im DALE-UV Verfahren erfolgt der technische Datenaustausch in Form von XML-Dokumenten. Für eine sinnvolle Vorgangsbearbeitung beim UV-Träger werden diese XML-Daten jedoch auf der Uni-DAV in eine bildhafte Darstellung (technisch PDF-Datei) gewandelt und zusammen mit dem XML-Dokument an den UV-Träger übermittelt.

Damit auch die Leistungserbringer (D-Ärzte) die Möglichkeit haben, ihre erfassten Daten in der gleichen bildhaften Form anzusehen wie die Sachbearbeitung beim UV-Träger, stellt die DGUV den Software-Herstellern der Krankenhaus- und Praxis-Systeme eine Softwarekomponente zur Verfügung, mit deren Hilfe die bildhafte Darstellung der DALE-UV XML-Dokumente erzeugt werden kann. Diese als „DALE-UV PDF-Generator“ bezeichnete Softwarekomponente ist als eigenständige, optional zu verwendende Anwendung zu betrachten. Sinnvollerweise wird das Werkzeug in die jeweiligen Produkte durch die Software-Hersteller integriert und an die Endanwender ausgeliefert. Ziel der Integration sollte sein, einem Anwender der Krankenhaus-/Praxissoftware nach Erfassung und Plausibilisierung seiner Berichtsdaten auf Knopfdruck die korrekte bildhafte Darstellung seines Berichts präsentieren zu können.

Der PDF-Generator ersetzt die bisher zu diesem Zweck verwendeten Stylesheets, die nicht mehr zentral durch die DGUV weitergepflegt werden.

Der PDF-Generator ist darüber hinaus in der Lage DALE-UV Quittungs-XML-Dateien, wie sie von der Datenannahmestelle an die Leistungserbringer versendet werden, in die bildhafte Darstellung zu konvertieren.

Dieses Dokument beschreibt die Installation und die Handhabung des DALE-UV PDF-Generators und richtet sich primär an technisch orientierte Mitarbeiter der Krankenhaus-/Praxissoftware-Hersteller.

Für Rückfragen zur Installation steht Ihnen das DALE-UV Support-Team telefonisch unter 02241 / 231-1330 oder per E-Mail [support@dale-uv.de](mailto:support@dale-uv.de) gern zur Verfügung.

### **Hinweis:**

Die vorliegende Version 6.0.\* des DALE-UV-PDF-Generators kann für die DALE-UV XML-Dokumente sowohl mit der Berichtsversion 19:4:01 als auch mit der Berichtsversion 22:2:01 eingesetzt werden.

## 3 Installation

### 3.1 Voraussetzungen

Bevor mit der Installation der PDF-Generator Softwarekomponente begonnen werden kann, sind einige Voraussetzungen zu prüfen und ggf. zu schaffen.

#### 3.1.1 Hardware-Voraussetzungen

Die Auslegung der Hardware zum Betrieb des PDF-Generators ist von der individuellen Verwendungshäufigkeit des Nutzers abhängig. Folgende Mindestkonfiguration sollte aber in der Regel ausreichend sein:

- Single-Prozessor-System (800 MHz)
- min. 256MB Hauptspeicher
- Platzbedarf Verzeichnisablage: ca. 35 KByte/Geschäftsvorfall (~35KB PDF)

#### 3.1.2 Software-Voraussetzungen

Der PDF-Generator wurde in Java entwickelt und ist mit einer Java-Virtual-Machine zu verwenden. Aus lizentechnischen Gründen wird keine Java VM mit ausgeliefert. Dadurch ist eine zusätzliche Installation einer Java VM zwingend notwendig, sofern nicht bereits eine geeignete Java VM auf Ihrem System installiert ist.

Um eine ordnungsgemäße Funktionalität zu gewährleisten, wird angeraten das quelloffene OpenJDK einzusetzen. Eine Übersicht, welche OpenJDK Version mit dem ausgelieferten PDF-Generator zu verwenden ist, finden Sie im Download-Bereich unter **Information zu kompatiblen OpenJDK-Versionen**.

Ausgeführt wird die Anwendung über einen Java-Konsolenbefehl und ist somit sowohl auf Windows als auch Linux-Systemen gleichermaßen ausführbar.

- **Adobe Reader**

Optional wird für die Betrachtung der PDF-Dokumente die Installation der frei verfügbaren Software „Adobe Reader“ empfohlen. Diese kann von <http://www.adobe.com> bezogen werden.

- **Unzip**

Die Installation setzt die Verfügbarkeit eines Werkzeugs zur Extraktion von ZIP-Dateien voraus (z.B. Winzip).

### 3.2 Installation PDF-Generator

Die Installation des PDF-Generators umfasst minimal das Entpacken eines ZIP-Archivs in einem lokalen Installationsverzeichnis.

#### 3.2.1 Grundinstallation

Die PDF-Generator Softwarekomponente finden Sie im Download-Bereich (siehe „Info für Softwarehersteller“ → „Downloads“), der über <http://www.dale-uv.de> erreichbar ist. Dort im

Bereich „Tools“ → „PDF-Generator“ unter der Bezeichnung „Softwarekomponente“. Die heruntergeladene Datei hat den Namen `daleuv2pdf_v6.0.0.zip`.

Extrahieren Sie den Inhalt des ZIP-Archivs in ein Zielverzeichnis Ihrer Wahl. Bei der Extraktion wird automatisch der Ordner `daleuv2pdf` als Installationsverzeichnis unterhalb des gewählten Zielverzeichnisses angelegt.

Das gewählte Installationsverzeichnis `daleuv2pdf` wird im weiteren Verlauf dieses Handbuchs als `[PDF_GENERATOR_PFAD]` bezeichnet.

***Hinweis:***

Wir empfehlen als Installationsziel (`[PDF_GENERATOR_PFAD]`) das Verzeichnis

`C:\daleuv2pdf`. Bei Linux-Systemen das Verzeichnis `/usr/daleuv2pdf`

Es kann aber ebenso ein anderes Installationsverzeichnis gewählt werden.

Unter Windows ab Version 7 und höher wird eine Installation in das Verzeichnis `C:\Programme` bzw. `C:\Programme (x86)` nicht empfohlen, da dieses eingeschränkten Schreibberechtigungen unterliegt.

## 4 Bedienung

Für die Verwendung des PDF-Generators ist es notwendig, dass Ihr System über eine funktionsfähige Java VM verfügt. Im folgenden Kapitel 4.1 wird exemplarisch für die Version `jdk-17.X.X` beschrieben, wie Sie eine neue Java VM in Ihrem System einrichten. Bei anderen Versionen sind die `jdk`-Versionsangaben inklusive der Unterversionsnummern „`X.X`“ anzupassen.

Falls Sie bevorzugen eine bestehende Java VM zu verwenden, können Sie Kapitel 4.1 überspringen und direkt zu Kapitel 4.2 übergehen.

### 4.1 Einrichten einer neuen Java VM

Um eine neue Java VM einzurichten müssen Sie sich zunächst für Ihr System den passenden Build herunterladen. Die Links zu den OpenJDK-Versionen finden Sie im Downloadbereich unter **Information zu kompatiblen OpenJDK-Versionen**.

Für Windows-Systeme erhalten sie ein ZIP-Archiv, für Linux Systeme analog ein Tar-Gz-Archiv.

Extrahieren Sie den Inhalt des Archivs in ein Zielverzeichnis Ihrer Wahl. Bei der Extraktion wird automatisch der Ordner `jdk-17.X.X` als Installationsverzeichnis unterhalb des gewählten Zielverzeichnisses angelegt.

**Hinweis:**

Auf Windows-Systemen empfehlen wir als Installationsziel (`[JAVA_HOME]`) das Verzeichnis `C:\jdk-17.X.X`

Unter Linux das Verzeichnis `/usr/java/jdk-17.X.X`

Das gewählte Installationsverzeichnis `jdk-17.X.X` wird im weiteren Verlauf dieses Handbuchs als `[JAVA_HOME]` bezeichnet.

Innerhalb des Ordners `jdk-17.X.X` befindet sich ein Unterordner `\bin` und innerhalb dieses Ordners wiederum befindet sich die ausführbare Java-Applikation. Unter Windows-Systemen heißt die ausführbare Datei `java.exe`, unter Linux fällt die Endung `.exe` weg.

Um zu überprüfen ob die Java VM lauffähig ist, kann man sie anweisen die Versionsnummer auszugeben. Siehe dazu Kapitel 4.3.

### 4.2 Verwenden einer bestehenden Java VM

Wenn Sie mithilfe des Kapitels 4.1 eine neue Java VM installiert haben können Sie dieses Unterkapitel überspringen und zu Kapitel 4.3 übergehen.

Sofern bereits eine Java VM installiert ist, ist es erforderlich den Pfad der Java-Applikation ausfindig zu machen. Unter Windows-Systemen heißt die Java-Applikation `java.exe` und befindet sich häufig unter `C:\Program Files\Java\jdk-17.X.X` bzw. `C:\Program Files (x86)\Java\jdk-17.X.X`

Bei Linux-Systemen fällt die Endung `.exe` weg. Gängige Pfade sind:

`/usr/lib/jvm/jdk-17.X.X` oder `/usr/java/jdk-17.X.X`

Haben Sie Ihre Java-Anwendung in Form einer `java.exe` bzw. unter Linux `java` ausfindig gemacht, dann können Sie überprüfen, ob die Java VM lauffähig ist, indem Sie die Java-Applikation anweisen die Versionsnummer auszugeben. Siehe dazu Kapitel 4.3.

### 4.3 Java VM Funktionalität prüfen

Um zu überprüfen ob die Java VM lauffähig ist, kann man sie anweisen die Versionsnummer auszugeben. Dafür muss über ein Konsolenfenster folgender Befehl abgesetzt werden:

```
[JAVA_HOME]\bin\java.exe -version (Windows)
```

bzw.

```
[JAVA_HOME]\bin\java -version (Linux)
```

#### **Hinweis:**

Wenn Sie als `[JAVA_HOME]`-Verzeichnis zum Beispiel `C:\jdk-17.X.X` gewählt haben, dann lautet ihr Kommandozeilenbefehl:

```
C:\jdk-17.X.X\bin\java.exe -version
```

Falls der Pfad Leerzeichen enthält, ist der Pfad in Anführungszeichen zu stellen, und würde demnach wie folgt lauten:

```
"C:\Pfad mit Leerzeichen\jdk-17.X.X\bin\java.exe" -version
```

Bei korrektem Vorgehen erscheint auf der Konsole eine Ausgabe folgender Art:

```
openjdk version "17.X.X" 2019-01-15
OpenJDK Runtime Environment XX.X (build 17.X.X+X)
OpenJDK XX-Bit Server VM 18.9 (build 17.X.X+X, mixed mode)
```

### 4.4 Ausführen des PDF-Generators

Der PDF-Generator wurde durch die Installation (Kapitel 3.2) in das Verzeichnis `[PDF_GENERATOR_PFAD]/daleuv2pdf_lib_v6.0.0.jar` extrahiert. Um das Programm – welches als JAR-Datei vorliegt - auszuführen muss über die Kommandozeile die Java VM mit dieser JAR-Datei als Parameter aufgerufen werden:

#### **Vor OpenJDK 18**

```
[JAVA_HOME]/java.exe
-jar [PDF_GENERATOR_PFAD]/daleuv2pdf_lib_v6.0.0.jar
```



## Ab OpenJDK 18

```
[JAVA_HOME]/java.exe -Dfile.encoding=[Encoding-Type]
-jar [PDF_GENERATOR_PFAD]/daleuv2pdf_lib_v6.0.0.jar
```

Es wird empfohlen als `Encoding-Type latin1` zu verwenden, da andernfalls die Darstellung von Umlauten und anderen Sonderzeichen im PDF von der Darstellung im XML abweichen könnte.

Nachdem die Platzhalter `[JAVA_HOME]` und `[PDF_GENERATOR_PFAD]` durch die absoluten Pfade ersetzt wurden sieht ein korrekter Befehl beispielsweise folgendermaßen aus:

### Windows-Konsole vor OpenJDK 18:

```
C:\jdk-17.0.2\bin\java.exe
-jar C:\daleuv2pdf\daleuv2pdf_lib_v6.0.0.jar
```

### Linux-Konsole vor OpenJDK 18:

```
/usr/java/jdk-17.0.2/bin/java
-jar /usr/daleuv2pdf/daleuv2pdf_lib_v6.0.0.jar
```

### Windows-Konsole ab OpenJDK 18:

```
C:\jdk-17.0.2\bin\java.exe -Dfile.encoding=latin1
-jar C:\daleuv2pdf\daleuv2pdf_lib_v6.0.0.jar
```

### Linux-Konsole ab OpenJDK 18:

```
/usr/java/jdk-17.0.2/bin/java -Dfile.encoding=latin1
-jar /usr/daleuv2pdf/daleuv2pdf_lib_v6.0.0.jar
```

#### **Hinweis:**

Wenn ein Pfad Leerzeichen enthält, muss er in Anführungszeichen stehen.

Z.B. "C:\Program Files\java\jdk-17.X.X\bin\java.exe" -jar [...]  
usw.

Bei dem gerade beschriebenen Befehl fehlen allerdings noch Parameter zur Steuerung der Anwendung. Wird der Befehl so abgesetzt wie oben beschrieben, kann keine PDF-Generierung stattfinden. Stattdessen gibt der PDF-Generator lediglich einige Informationen zusammen mit einer Liste der möglichen Parameter aus:

```
*****
Titel: DALE-UV PDF-Generator
Organisation: DGUV
Anwendung: Erzeugung DALE-UV PDF aus DALE-UV XML
Version: 6.0.0
JRE-Version: 17.0.2
*****

Aufruf: [PDF_GENERATOR_APP] -i <Pfad>/Dateiname[.xml] -o <Ausgabepfad> [-g] [-w] [-s]

-i: Pfad und Dateiname des DALE-UV XML-Dokuments
-o: Verzeichnis, in dem die PDF-Datei erzeugt werden soll
-g: Erzeugung der Berichtsversion fuer die gesetzl. Krankenversicherung
-w: Erzeugung der Berichtsversion fuer den weiterbeh. Arzt / Konsiliararzt
-s: keine Erzeugung der Standardberichtsversion (UV-Traeger)
```

Sie sollten den Befehl zu Testzwecken einmal in Ihrem System ausführen und Ihren PDF-Generator solch eine Ausgabe erzeugen lassen, bevor Sie mit dem nächsten Kapitel fortfahren.

Um das nun folgende Kapitel schlanker zu halten wird der Befehl zum Ausführen des PDF-Generators im weiteren Verlauf dieses Handbuchs als [PDF\_GENERATOR\_APP] bezeichnet.

## 4.5 Aufruf-Syntax

Der PDF-Generator ist eine konsolenorientierte Anwendung und besitzt keine eigene graphische Oberfläche. Die Anwendung wird allein durch – an den Befehl angehängte – Parameter gesteuert. Es muss zwingend folgende Aufruf-Syntax eingehalten werden:

```
[PDF_GENERATOR_APP] -i <Pfad>/Dateiname[.xml] -o <Ausgabepfad> [-g] [-w] [-s]
```

Parameter	Beschreibung	Hinweis	Angabe
-i	Parameter für die XML-Eingangsdatei	Die Pfadangabe muss auf eine vorhandene DALE-UV XML-Datei verweisen.	muss
-o	Parameter für den Ausgabepfad	In diesem Verz. wird/werden die PDF-Datei(en) erzeugt. Der Ausgabepfad muss bereits existieren. Vorhandene Dateien mit demselben Ausgabedateinamen werden überschrieben.	muss
[-g]	Zusätzliche Erzeugung des Berichtsexemplars für die Gesetzliche Krankenversicherung	Dies gilt nur für die Nachrichtentypen DABE und VERB. Bei allen anderen Nachrichtentypen wird die Option, wenn sie eingeschaltet ist, ignoriert.	kann
[-w]	Zusätzliche Erzeugung des Berichtsexemplars für den weiterbehandelnden Arzt / Konsiliararzt	Dies gilt nur für die Nachrichtentypen DABE und VERB. Bei allen anderen Nachrichtentypen wird die Option, wenn sie eingeschaltet ist, ignoriert.	kann
[-s]	Keine Erzeugung des Standardberichtsexemplars	Über diesen Schalter kann explizit die Erzeugung des Standardberichtsexemplars (Bericht an den UV-Träger) deaktiviert werden; wenn z.B. nur das Exemplar für die Krankenkasse und/oder weiterbehandelnden Arzt erzeugt werden soll; die Option ist nur in Verbindung mit [-g] und/oder [-w] zulässig	kann

Tabelle 2: daleuv2pdf\_lib\_v6.0.0.jar Parameter

### **Hinweis:**

Es wird nicht überprüft, ob die Eingangsdatei aus fachlicher Sicht DALE-UV XML konform ist. Der Nutzer liegt hier in der Verantwortung, dies sicherzustellen (z.B. mit Hilfe des ebenfalls durch die DGUV bereitgestellten Plausi-Moduls).

Darüber hinaus muss sichergestellt werden, dass das Output-Verzeichnis vorhanden ist.

## (1) Der Beispielaufruf

```
C:\jdk-17.0.2\bin\java.exe -Dfile.encoding=latin1
-jar C:\daleuv2pdf\daleuv2pdf_lib_v6.0.0.jar
-i C:\daleuv2pdf\test\234567890_17_9900953_01.xml
-o C:\daleuv2pdf -g -w
```

erzeugt im Verzeichnis `c:\daleuvpdf` folgende PDF-Dateien (Annahme, die Eingangsdatei `234567890_17_9900953_01.xml` ist ein DABE oder VERB, da nur diese Berichtstypen an die Gesetzliche Krankenversicherung und den weiterbehandelnden Arzt versendet werden):

```
234567890_17_9900953_01.pdf
234567890_17_9900953_01_GKV.pdf
234567890_17_9900953_01_WBA.pdf
```

## (2) Der Beispielaufruf

```
C:\jdk-17.0.2\bin\java.exe -Dfile.encoding=latin1
-jar C:\daleuv2pdf\daleuv2pdf_lib_v6.0.0.jar
-i C:\daleuv2pdf\test\234567890_17_9900953_01.xml
-o C:\daleuv2pdf
```

erzeugt im Verzeichnis `c:\daleuvpdf` die PDF-Datei mit Namen:

```
234567890_17_9900953_01.pdf .
```

## 4.6 Rückgabewerte des Aufrufs

Generell liefert das Programm `daleuv2pdf_lib_v6.0.0.jar` definierte Rückgabewerte (Exit-Codes). Ein Rückgabewert ungleich 0 bedeutet dabei, dass innerhalb der Verarbeitung ein Fehler aufgetreten ist. Die konkrete Fehlermeldung wird vom Programm auf die Standard-Fehlerausgabe (`stderr`) geschrieben. Bei der Integration des Werkzeugs in ein Krankenhaus/Praxis-System sollte daher nach Rückkehr des Aufrufs stets der Rückgabewert abgefragt und ausgewertet werden.

## 5 Test des PDF-Generators

Für den Test der Installation steht der Ordner [PDF\_GENERATOR\_PFAD]\test mit einem Beispielbericht in Form einer Nutzdatsatzdatei zur Verfügung:

234567890\_17\_9900953\_01.xml (D-Arztbericht der Version 22:2:01 der technischen Dokumentation DALE-UV)

Für einen Installationstest gehen Sie wie folgt vor:

1. Öffnen Sie eine Windows-Kommando-Shell („Start“ → „Ausführen“; Befehl „cmd“ eingeben)
2. Wechseln Sie in das Test-Programmverzeichnis (wobei Sie [PDF\_GENERATOR\_PFAD] durch den tatsächlichen Installationspfad ersetzen):  
cd [PDF\_GENERATOR\_PFAD]\test
3. Tippen Sie folgenden Befehl

```
[PDF_GENERATOR_APP] -i
[PDF_GENERATOR_PFAD]\test\234567890_17_9900953_01.xml -o
[PDF_GENERATOR_PFAD]\test
```

Sie erhalten daraufhin folgende Konsolenausgaben:

```
c:\daleuv2pdf>c:\java\openjdk-17.0.2\bin\java -jar daleuv2pdf_lib_v6.0.0.jar -i test\
234567890_17_9900953_01.xml -o test\
*****
Titel: DALE-UV PDF-Generator
Organisation: DGUV
Anwendung: Erzeugung DALE-UV PDF aus DALE-UV XML
Version: 6.0.0
JRE-Version: 17.0.2
*****
PDF-Datei erzeugt: c:\daleuv2pdf\test\234567890_17_9900953_01.pdf
```

Im Verzeichnis [PDF\_GENERATOR\_PFAD]\test sollte nun die Datei 234567890\_17\_9900953\_01.pdf liegen.

Analog können Sie mit den Optionsschaltern [-g] und [-w] verfahren:

1. Schritte 1 und 2 wie im vorherigen Test
2. Löschen Sie evtl. die im vorherigen Test erstellte 234567890\_17\_9900953\_01.pdf Datei.
3. Tippen Sie folgenden Befehl

```
[PDF_GENERATOR_APP] -i
[PDF_GENERATOR_PFAD]\test\234567890_17_9900953_01.xml -o
[PDF_GENERATOR_PFAD]\test -g -w
```

Sie erhalten daraufhin folgende Konsolenausgaben:

```
c:\daleuv2pdf>c:\java\openjdk-17.0.2\bin\java -jar daleuv2pdf_lib_v6.0.0.jar -i test\
234567890_17_9900953_01.xml -o test\ -g -w
*****
Titel: DALE-UV PDF-Generator
Organisation: DGUV
Anwendung: Erzeugung DALE-UV PDF aus DALE-UV XML
Version: 6.0.0
JRE-Version: 17.0.2
*****
PDF-Datei erzeugt: c:\daleuv2pdf\test\234567890_17_9900953_01.pdf
PDF-Datei erzeugt: c:\daleuv2pdf\test\234567890_17_9900953_01_GKV.pdf
PDF-Datei erzeugt: c:\daleuv2pdf\test\234567890_17_9900953_01_WBA.pdf
```

Im Verzeichnis [PDF\_GENERATOR\_PFAD]\test sollten nun die Dateien

```
234567890_17_9900953_01.pdf
234567890_17_9900953_01_WBA.pdf
234567890_17_9900953_01_GKV.pdf
```

liegen.

## 6 Java-Schnittstelle

In den vorangegangenen Kapiteln wurde beschrieben wie die PDF-Generator Applikation über die Kommandozeile ausgeführt werden kann. Es ist allerdings auch möglich den PDF-Generator direkt in ein Java-Programm einzubinden.

### **Hinweis:**

Die Java-Schnittstelle setzt mindestens Java Version 8 voraus.

### 6.1 Beschreibung

Um die Java-Schnittstelle des PDF-Generators in ein Java-Programm einzubinden, muss die Bibliothek `daleuv2pdf_lib_v6.0.0.jar` in den Java-Klassenpfad der Applikation eingebunden werden. In dieser JAR-Datei ist die Klasse `Dale2PdfInvokerFactory` enthalten, über deren statische Methode `createDale2PdfInvoker()` eine Instanz des `Dale2PdfInvoker` erzeugt werden kann. Mit dieser Instanz lassen sich unter Angabe eines Unfallberichtes im XML-Format und eines Ausgabepfades (jeweils der absolute Systempfad) die Berichte im PDF-Format generieren.

Zu verwenden ist hierfür die Standardmethode `createPdf(String, String)`, bzw. die zusätzlichen Methoden `createPdfGkv(String, String)` und `createPdfWba(String, String)` für die Generierung von Berichten mit eingeschränkter Sichtbarkeit für Gesetzliche Krankenkassen oder weiterbehandelnde Ärzte. Diese Methoden liefern keinen Rückgabewert, werfen bei einem Fehler aber eine Exception, die explizit zu behandeln ist.

Die zu verwendenden Klassen (`Dale2PdfInvoker`, `Dale2PdfInvokerFactory`) befinden sich im Package `dguv.daleuv.report.client`.

### 6.2 Beispiel-Listing

```
// Instanz des PDF-Invokers erzeugen
Dale2PdfInvoker pfdInvoker = Dale2PdfInvokerFactory.createDale2PdfInvoker();
try
{
    // UVT-Bericht in c:\tmp erzeugen lassen (für Unfallversicherung)
    pfdInvoker.createPdf("c:/daleuv2pdf/test/234567890_17_9900953_01.xml", "c:/tmp");
    // GKV-Bericht in c:\tmp erzeugen lassen (für Krankenversicherung)
    pfdInvoker.createPdfGkv("c:/daleuv2pdf/test/234567890_17_9900953_01.xml", "c:/tmp");
    // WBA-Bericht in c:\tmp erzeugen lassen (für weiterbehandelnden Arzt)
    pfdInvoker.createPdfWba("c:/daleuv2pdf/test/234567890_17_9900953_01.xml", "c:/tmp");
}
catch (Exception e)
{
    // Fehlermeldung aus Exception ermitteln und auf stdout ausgeben
    System.out.println(e.getMessage());
}
```

## 7 Log-Ausgabe umkonfigurieren (Log4j)

Für das Loggen von Anwendungsmeldungen innerhalb der PDF-Generator Softwarekomponente kommt das Logging-Framework *log4j* zum Einsatz.

Dieses Logging-Framework kann über eine Konfigurationsdatei (den s.g. *log4j.properties*) konfiguriert werden. Das Java Archiv *daleuv2pdf\_lib\_v6.0.0.jar* beinhaltet eine solche Konfigurationsdatei, die per default herangezogen wird. Dort ist unter anderem konfiguriert, dass Anwendungsmeldungen auf der Standard-Ausgabe des ausführenden Systems ausgegeben werden.

Für die Einbettung der PDF-Generator Softwarekomponente in das eigene Softwaresystem ist es möglich, die *log4j.properties* Ihren Erfordernissen entsprechend anzupassen. Beispielsweise um mittels s.g. *Appender* die Ausgabeziele der erzeugten Loggingausgaben festzulegen.

Der einfachste Weg eigene *log4j.properties* zu verwenden ist, den Pfad zu den *eigenen* *log4j.properties* per Java-Parameter mitzugeben.

Beispielaufruf:

```
c:\jdk-17.0.2\bin\java.exe -Dfile.encoding=latin1
-Dlog4j.configuration=file:C:\daleuv2pdf\log4j.properties
-jar C:\daleuv2pdf\daleuv2pdf_lib_v6.0.0.jar
-i C:\daleuv2pdf\test\234567890_17_9900953_01.xml
-o C:\daleuv2pdf\test
```

In Falle des Beispiels müsste sich die Datei *log4j.properties* unter *C:\daleuv2pdf* befinden. Wo Sie Ihre *log4j.properties* letztlich ablegen, bleibt Ihnen überlassen.

Sollte der Pfad zu den *log4j.properties* nicht korrekt sein, wird dies durch eine Fehlermeldung wie z.B. der nachfolgenden angezeigt. Die Angaben des Pfads oder die Ablage der *log4j.properties* im korrekten Verzeichnis wären anzupassen.

```
log4j:ERROR Could not read configuration file from URL
[file:C:/daleuv2pdf/log4j.properties].
java.io.FileNotFoundException: C:\daleuv2pdf\log4j.properties (Das
System kann die angegebene Datei nicht finden)
[...]
log4j:ERROR Ignoring configuration file
[file:C:/daleuv2pdf/log4j.properties].
```