

Produkthandbuch Kernprüfung

Version 2.5.0

Deutsche Rentenversicherung Bund

Kernprüfprogramme KP

Inhaltsverzeichnis

1. Einleitung ... 4
2. Programm beziehen ... 5
3. Komponenten ... 5
3.1 „Kernprüfung Routing Programm der SV“ (KRIPS) ... 6
3.2 Regelwerk und einzelne Datenstruktur-Kernprüfungen ... 7
3.3 Komponentenverteilung in Jar-Archive ... 8
4. Drittabhängigkeiten ... 8
4.1 JAXB-API ... 10
5. Installation ... 10
5.1 Systemvoraussetzungen ... 10
5.2 Klassenpfad ... 10
5.3 Multithreading ... 10
5.4 Aufrufmöglichkeiten ... 11
5.4.1 Allgemeine Schnittstelle einer Datenstruktur-Kernprüfung ... 11
5.4.2 Beispiel - Indirekter Aufruf mittels KRIPS Komponente ... 11
5.4.3 Beispiel - Direkter Aufruf der Datenstruktur Kernprüfung ... 12
5.4.4 Aufrufparameter - Kernprüfung mit fester Datenstruktur ... 12
5.4.5 Aufrufparameter - XML-Kernprüfung ... 13
6. Logging ... 14
7. Schemavalidierung bei XML-Datenstruktur-Kernprüfungen ... 15
8. Versionierung ... 15
9. Haftungsausschluss ... 16
10. Anhang: Programmdokumentation ... 16
10.1 Aufbau der einzelnen Kernprüfungen ... 16
10.1.1 Datenstruktur-Kernprüfungen Implementierung ... 16
10.1.2 Konfiguration der Struktur ... 17
10.1.3 Konfiguration der auszuführenden Prüfungen ... 17
10.1.4 Fehlertexte ... 18
10.1.5 Allgemeine Informationen ... 18

10.2 Kernprüfung Basis 19
10.2.1 kernpruefung-interface 19
10.2.2 kernpruefung-basis-technisch 19
10.2.3 kernpruefung-basis-fachlich 19
10.2.4 datenaufbereitung 20
11. Anlage: Lizenztexte 20
11.1 Apache 2.0 20
11.2 MPL 2.0 28
11.3 EDL 1.0 43
11.4 MIT 44
11.5 BSD-2-Clause 44
12. Anlage: Änderungshistorie 45

1. Einleitung

Die Kernprüfung ist eine flexible in Java implementierte modulare Prüfbibliothek, welche sehr vielseitig einsatzbar ist und somit sehr einfach – sowohl in Java-SE- als auch Java-EE-basierte Anwendungen – eingebunden und verwendet werden kann. Dabei ermöglicht die Kernprüfung sowohl die Überprüfung von XML-basierten Datensätzen als auch die Prüfung von Datensätzen mit fixen Längen.

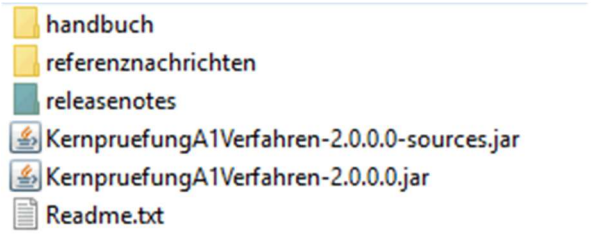
Bei der Konzeption und Umsetzung der Kernprüfung wurde vor allem auf folgende wichtige Eigenschaften Wert gelegt:

- **Hohe Wiederverwendbarkeit**: Dies wird zum einen dadurch erreich, dass alle Kernprüfungen eine gemeinsame technische Basis besitzen. Zum anderen wurden in einer aufwendigen Analyse fachlich identische Prüfungen für Felder und Bausteine identifiziert und in entsprechenden gemeinsamen Bibliotheken, der fachlichen Basis, ausgelagert.
- **Einfache Implementierung**: Dies wird vor allem dadurch gewährleistet, dass die Prüfungen für die einzelnen Felder, Bausteine oder Datensätze in der Regel unter Verwendung von den bereitgestellten technischen Basisprüfungen konfiguriert werden. Des Weiteren gibt es für immer wiederkehrende Felder und Bausteine vordefinierte, anpassbare Konfigurationen, die übernommen werden können.
- **Konsistenz**: Dieser Aspekt ergibt sich vor allem als Folge der Wiederverwendung, denn auf diese Weise ist sichergestellt, dass jede Kernprüfung garantiert die gleichen Prüfungen durchführt, wenn beispielsweise eine Versicherungsnummer validiert werden soll. Das Resultat der Überprüfung ist bei gleichen Eingabedaten somit auch bei Wiederholung identisch.
- **Gute Erweiterbarkeit**: Anpassungen, wie beispielsweise die Hinzunahme von gesonderten Absendernummern, können an einer zentralen Stelle einmalig implementiert und anschließend verteilt werden.
- **Reduzierung des Testaufwands**: Da sowohl die Prüfungen der technischen Basis als auch die gemeinsamen fachlichen Prüfungen umfangreich getestet wurden, sind hierzu durch die Entwicklung keine detaillierten Tests notwendig. Dies reduziert den Entwicklungsaufwand gemäß unserer Erfahrung erheblich.
- **Gute Wartbarkeit**: Dies wird unter anderem durch die Aufteilung der kompletten Kernprüfungsinfrastruktur in mehrere, einzelne Module erreicht. Damit ergibt sich eine erhöhte Wartbarkeit und schnellere Reaktionszeit, da bei möglichen Anpassungen nur kleine Einheiten ausgetauscht werden müssen und nicht die komplette Kernprüfung betroffen ist.

Ziel dieses Dokumentes ist es einen groben Überblick über die Module der Kernprüfung, sowie Tipps für die korrekte Installation zu geben.

2. Programm beziehen

Die Kernprüfung kann über den Internetauftritt der Deutschen Rentenversicherung bezogen werden.[1] Außerdem stellt der GKV-Spitzenverband die Kernprüfung für die Krankenkassen bereit. Für unterschiedliche Verfahren werden hier einzelne Programmpakete zum Download zur Verfügung gestellt. Ein Verfahrenspaket beinhaltet die folgenden Bestandteile (hier am Beispiel der Kernprüfung zum Verfahren A1 dargestellt):



Im Ordner „handbuch“ befindet sich die aktuelle Dokumentation zur Verwendung der Kernprüfung. Einige Beispieldatensätze zum Verfahren befinden sich in dem Verzeichnis „referenznachrichten“ und können als Testfälle verwendet werden. Mithilfe dieser Datensätze kann überprüft werden, ob die Kernprüfung korrekt installiert wurde. Des Weiteren befinden sich in der Lieferung die folgenden Module, welche jeweils zusätzlich mit Quelltext vorliegen:

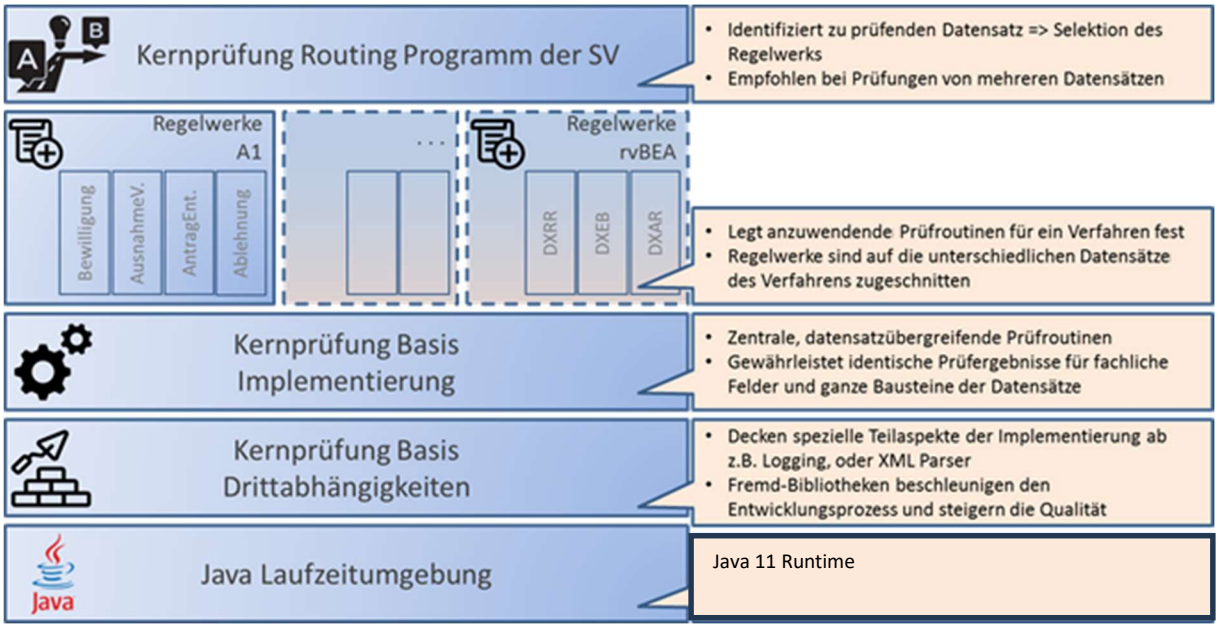
- Kernpruefung<*NameDesFachverfahrens*>-x.x.x.jar

Das Jar-Archiv wird in Kapitel 3 hinsichtlich der enthaltenen Komponenten und in Kapitel 3.3 bezüglich deren Verteilung auf die genannten Jar-Archive beschrieben.

3. Komponenten

Die Kernprüfung besteht aus mehreren Anwendungsschichten, die aufeinander aufbauen und sich zur Gesamtapplikation ergänzen.

[1] www.dsrv.info/de/Navigation/20_Unsere_Verfahren/01_Nationaler_Datenaustausch/09_Kernprüfung/Kernprüfung_node.html



Das **„Kernprüfung Routing Programm der SV“ (KRIPS)** leitet den Aufruf an das Regelwerk einer **konkreten Kernprüfung** weiter. Alternativ ist es auch möglich auf die Nutzung der KRIPS Komponente zu verzichten. Wird keine KRIPS Komponente verwendet, muss das Regelwerk einer konkreten Kernprüfung direkt angesprochen werden. Das Regelwerk nutzt seinerseits die Komponenten der **Kernprüfung Basis** Implementierung Anwendungsschicht. Diese Anwendungsschichtsetzt sich aus der **fachlichen Basis** für gemeinsame Feld- und Bausteinprüfungen und der **technischen Basis** für gemeinsame technische Prüfungen sowie zur gemeinsamen Ablauflogik zusammen. Die Kernprüfung Basis verwendet dabei **Drittabhängigkeiten**, die spezielle Aspekte der Gesamtimplementierung abdecken.

3.1 „Kernprüfung Routing Programm der SV“ (KRIPS)

Mit dem „KRIPS“ wird eine zentrale Aufrufschnittstelle aller einzelnen Datenstruktur-Kernprüfungen zur Verfügung gestellt. Dabei leitet er alle in ihm registrierten Datenstrukturen an die entsprechende Datenstruktur-Kernprüfung weiter. Unbekannte Datenstrukturen werden von ihm mit einer qualifizierten Fehlermeldung abgewiesen. Somit ist sichergestellt, dass mit dem KRIPS nur bekannte Datenstrukturen verarbeitet werden können.

Der Einsatz dieses Moduls wird dringend empfohlen, wenn die zur Prüfung eingehenden Daten aus unterschiedlichen Datensätzen bestehen. Durch das KRIPS Modul wird die Aufrufentscheidung welche Datenstruktur mit welchem Kernprüfungs-Regelwerk behandelt wird in einer zentralen Stelle realisiert und muss nicht individuell implementiert werden. Alternativ kann die individuelle Kernprüfung direkt aufgerufen werden.

Der KRIPS dient nicht zur Steuerung des Aufrufs verschiedener aktueller und vorheriger Versionen der Kernprüfung.

3.2 Regelwerk und einzelne Datenstruktur-Kernprüfungen

Die verschiedenen Fachverfahren (z.B. A1, DEÜV, rvBEA und weitere) arbeiten mit einer oder mehreren Datenstrukturen, deren Validierung in sogenannten Regelwerken beschrieben wird. Ein Regelwerk beinhaltet dabei die notwendigen einzelnen Datenstruktur-Kernprüfungen. Selbige werden vom KRIPS oder direkt ausgeführt.

Aktuell werden die folgenden Datenstruktur-Kernprüfungen angeboten:

A1	Stand: 03.08.2026	kernpruefung-a1antragentsendung kernpruefung-a1ausnahmevereinbarungarbeitgeber kernpruefung-a1bewilligung kernpruefung-a1ablehnung kernpruefung-a1antragbeamte kernpruefung- a1antraggmeeinarbeitgeber kernpruefung- a1antragflugundkabinenbesatzung kernpruefung- a1antragseeleute kernpruefung-a1antragantragselbstaendige kernpruefung-a1antraggmeglobal kernpruefung-antraggrenzgaenger kernpruefung-a1ausnahmevereinbarungerwerbstaetigeundrentner
SVA	Stand: 03.08.2026	kernpruefung-svaablehnung kernpruefung-svaantragav kernpruefung-svaantragentsendung kernpruefung-svaantragentsendungselbststaendige kernpruefung-svaausnahmevereinbarungselbststaendige kernpruefung-svagenehmigung
rvBEA	Stand: 03.08.2026	kernpruefung-dxrr (Hinweis: Der DXRR ist derzeit nur in der Testumgebung zulässig! Der Registrierungsdatensatz wird für das automatisierte Testverfahren rvBEA genutzt. Eine Beschreibung zum automatisierten Test finden sie unter www.rvbea.de im Downloadbereich.) kernpruefung-dxeb kernpruefung-dxar
DEUEV	Stand: 03.08.2026	kernpruefung-dsme kernpruefung-dsae kernpruefung-dsbd kernpruefung-dsvv kernpruefung-dsak kernpruefung-dsfz

BNA	Stand: 03.08.2026	kernpruefung-bw02
GKV-Mon	Stand: 03.08.2026	kernpruefung-dskk

Von diesen Datenstruktur-Kernprüfungen sind natürlich nur diejenigen einzubinden, die auch im konkreten Anwendungsfall benötigen werden.

Zusätzliche Informationen und eine detaillierte Beschreibung der Bestandteile dieser Prüfungen befinden sich im Anhang.

3.3 Komponentenverteilung in Jar-Archive

Alle Komponenten der beschriebenen Anwendungsschichten, werden in verschiedenen Jar-Archiven zusammengefasst.

Die Regelwerke zu einem Fachverfahren bilden jeweils ein eigenes Jar-Archiv mit der Namenskonvention Kernpruefung<*NameDesFachverfahrens*>Verfahren-.x.x.x.jar. Um dieses Archiv erstellen zu können, werden mehrere einzelne Jar-Archive der Regelwerke der konkreten Datenstrukturen zusammengepackt. Dieses Format eines Jar-Archivs wird im weiteren Verlauf als SingleJar bezeichnet. Entsprechend der Beschreibung in Kapitel 3.2 werden alle Datenstrukturen eines Fachverfahrens in einem eigenen SingleJar gebündelt. Außerdem werden diesem Archiv die Jar-Archive der Komponenten Kernprüfung Basis Implementierung und Kernprüfung Basis Drittabhängigkeiten hinzugefügt. Beispielsweise werden dem Jar-Archiv KernpruefungA1Verfahren-x.x.x.jar die Datenstrukturen A1 Antrag Entsendung, A1 Ausnahmevereinbarung, A1 Bewilligung und A1 Ablehnung zugeordnet.

4. Drittabhängigkeiten

Der Einsatz von wiederverwendbaren Komponenten und Bibliotheken ist in der modernen Software-Entwicklung weit verbreitet. Dahinter steht der Gedanke, dass mit dem Einsatz von wiederverwendbaren Komponenten und Bibliotheken sowohl der initiale Entwicklungs- und Testaufwand als auch der Aufwand für Pflege und Weiterentwicklung erheblich reduziert werden kann. Die große Entwicklergemeinschaft, die sich an der Erstellung und kontinuierlichen Pflege von OpenSource Bibliotheken beteiligt, gewährleisten typischerweise auch eine ausgezeichnete Qualität und Domänen-spezifisches Wissen.

Aus diesem Grund wurde, wie zuvor erwähnt, auch bei der Entwicklung der Kernprüfung sehr viel Wert auf Wiederverwendbarkeit gelegt. Darüber hinaus gibt es mittlerweile von verschiedenen Herstellern und Vereinigungen eine Reihe von Bibliotheken, mit denen sich immer wieder kehrende Anforderungen und Probleme recht einfach und robust umsetzen bzw. lösen lassen. Oft ist davon abzuraten die in gängigen Bibliotheken bereits vorhandenen Funktionalitäten im Rahmen eines Projekts neu zu entwickeln, da diese selbst entwickelten Lösungen oft erhebliche Einschränkungen besitzen, deren Problematik oft erst im Verlauf eines Projektes erkennbar sind und die in der Folge

relativ umfangreichen Anpassungs- und Pflegeaufwand bedeuten. Dies haben auch Hersteller von Produkten (IBM WebSphere Application Server, RedHat JBoss AS) oder Frameworks (beispielsweise Spring) erkannt und nicht zuletzt ist deshalb die Verwendung einiger dieser Bibliotheken bei der Erstellung von Java-Anwendungen mittlerweile Standard. Die gängigen OpenSource Software Lizenz-Bedingungen erlauben zudem die Einbindung und Verteilung der Bibliotheken in eigenen Anwendungen ohne Gebühren.

Aus diesem Grund setzt auch die Kernprüfung ihrerseits gängige Bibliotheken ein, die im Folgenden genauer beschrieben werden.

Es wurde sich entschieden die Kernprüfung konfigurierbar zu machen. Dabei wurde der weit verbreitete XML-Standard verwendet. Für diese Funktionalität und das Einlesen der Konfigurationsdateien werden folgenden Bibliotheken verwendet:

Bibliothek	Lizenz	Quelle
commons-io-2.22.0.jar	Apache 2.0	https://commons.apache.org/proper/commons-io/
Saxon-HE-12.9jar	MPL 2.0	https://www.saxonica.com/download/java.xml
commons-pool2-2.13.1.jar	Apache 2.0	https://commons.apache.org/proper/commons-pool/
jakarta.activation-api-2.1.3.jar	EDL 1.0	https://jakarta.ee/specifications/xml-binding/4.0/
jakarta xml bind-api-4.0.2.jar	EDL 1.0	https://jakarta.ee/specifications/xml-binding/4.0/
aalto-xml 1.3.0	Apache 2.0	https://github.com/FasterXML/aalto-xml/blob/main/LICENSE
log4j-api 2.17.1	Apache 2.0	https://logging.apache.org/log4j/2.12.x/license.html
slf4j-api 1.7.25	MIT	https://www.slf4j.org/license.html
stax2-api 4.2	BSD-2-Clause	https://github.com/FasterXML/stax2-api/blob/main/LICENSE
xmlresolver 5.3.3	Apache 2.0	https://codeberg.org/XmlResolver/xmlresolver/src/branch/main/LICENSE.md

Für die wiederverwendbaren Prüfroutinen werden folgende Bibliotheken benötigt:

Bibliothek	Lizenz	Quelle
commons-lang3-3.20.0jar	Apache 2.0	https://commons.apache.org/proper/commons-lang/

Inhaber der Apache 2.0 Lizenz ist: Apache Software Foundation

Inhaber der MPL 2.0 Lizenz ist: Mozilla Foundation

Inhaber der EDL 1.0 Lizenz ist: Eclipse Foundation

Alle Bibliotheken befinden sich im Verzeichnis „lib“ des zur Verfügung gestellten Archivs. Bitte nehmen Sie alle Bibliotheken in den Klassenpfad auf.

4.1 JAXB-API

Mit der Kernprüfung wird nur die JAXB-API ausgeliefert. Damit diese verwendet werden kann, ist eine zusätzliche Runtime-Bibliothek erforderlich, die durch den Anwender selbst eingebunden werden muss.

Bei der DSRV wurde in der Entwicklung und bei den Tests die "org.glassfish.jaxb:jaxb-runtime:4.0.5" benutzt.

5. Installation

5.1 Systemvoraussetzungen

Die Kernprüfung wurde mit Java SE in der Version 8 implementiert. Voraussetzung ist also eine JRE Version 11.

5.2 Klassenpfad

In dem entsprechenden Kernprüfungs-Archiv finden Sie mehrere Verzeichnisse, die die nötigen Komponenten der Kernprüfungen je nach Abhängigkeit zusammenhalten. Fügen Sie bitte alle Bibliotheken dem Klassenpfad hinzu.

5.3 Multithreading

Bei der Entwicklung der Kernprüfung wurde darauf geachtet, dass diese multithreadingfähig und damit auch parallelisierbar ist. Aus diesem Grund wurden auf statische Instanzvariablen und Pattern wie dem Singleton bewusst verzichtet. Dennoch hat die Kernprüfung Objekte, die über die gesamte Lebensdauer erhalten bleiben. Aus diesem Grund ist es notwendig für jeden Thread eine eigene Instanz der Kernprüfung zu erstellen.

5.4 Aufrufmöglichkeiten

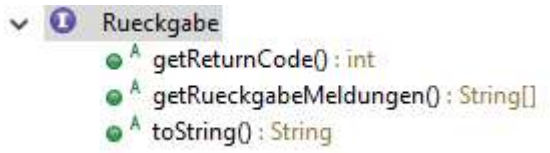
Je nach Nutzung der KRIPS Komponente in der Aufrufhierarchie, siehe Kapitel 3.3, ergeben sich verschiedene Aufrufmöglichkeiten, die in den folgenden Kapiteln detailliert erläutert werden. Neben einem direkten Aufruf des Regelwerks eines Fachverfahrens können auch einzelne Datenstruktur-Kernprüfungen aufgerufen werden. Die KRIPS Komponente ermöglich darüber hinaus die Verwendung mehrerer Fachverfahren mittels einer zentralen Schnittstelle.

5.4.1 Allgemeine Schnittstelle einer Datenstruktur-Kernprüfung

Jede Datenstruktur-Kernprüfung und der KRIPS implementieren das Interface kernpruefung.Kernpruefung. Dieses hat genau eine Methode:

```
public Rueckgabe pruefe(String datensatz, String vorlaufsatz);
```

Der Kernprüfung werden die Datenstruktur (feste Struktur oder XML) und der Vorlaufsatz übergeben. Die Rueckgabe hat folgende Methoden:



Der ReturnCode kann folgende Werte annehmen:

ReturnCode	Beschreibung	RückgabeMeldungen verfügbar?
0	keine Fehler, erfolgreiche Prüfung	
1	es gibt Hinweise	Ja
2	es gibt Fehler	Ja
3	es gibt Hinweise und Fehler	Ja
4	technischer Fehler	
5	Fehler bei der Verarbeitung im KRIPS	Ja
6	Fehler bei der Schemavalidierung (nur XML-Kernprüfungen)	

5.4.2 Beispiel - Indirekter Aufruf mittels KRIPS Komponente

Alle Kernprüfungen können zentral über den KRIPS folgendermaßen aufgerufen werden:

```
String datensatz = "...";
String vorlaufsatz = "...";

Kernpruefung krips = new Krips();
Rueckgabe rueckgabe = krips.pruefe(datensatz, vorlaufsatz);

System.out.println(rueckgabe.getReturnCode());

for (String rm : rueckgabe.getRueckgabeMeldungen()) {
        System.out.println(rm);
}
```

In Kapitel 5.4.4 und Kapitel 5.4.5 werden detaillierte Beispiele für die unterschiedlichen zu übergebenden Datentypen bereitgestellt.

5.4.3 Beispiel - Direkter Aufruf der Datenstruktur Kernprüfung

Wird die KRIPS Komponente nicht gesondert als Jar-Archiv verwendet, kann die Datenstruktur-Kernprüfung auch direkt angesprochen werden. Für jedes Fachverfahren wird bei dieser Aufrufmöglichkeit namentlich eine andere Klasse verwendet.

Im folgenden Beispiel wird die Datenstruktur-Kernprüfung des DSME aus dem Fachverfahrens DEÜV verwendet:

```
String datensatz = "";
String vorlaufsatz = "";

Kernpruefung kernpruefungDsme = new KernpruefungDsme();
Rueckgabe rueckgabe = kernpruefungDsme.pruefe(datensatz, vorlaufsatz);

System.out.println(rueckgabe.getReturnCode());

for (String rm : rueckgabe.getRueckgabeMeldungen()) {
        System.out.println(rm);
}
```

5.4.4 Aufrufparameter - Kernprüfung mit fester Datenstruktur

Bei Kernprüfungen fester Datenstrukturen muss als Datensatz ein vollständiger, einzelner Datensatz entsprechend der Fachvorgabe übergeben werden. Als Vorlaufsatz wird hier ein vollständiger zur Datenstruktur passender Vorlaufsatz erwartet.

```
String datensatz = " DSAEDEUEV90235319      66667777
011999010112315501010100064180251H5060A15035218      89R 013
JN              DBAZJ521997010119970331 ";
String vorlaufsatz = " VOSZKVTRV15451439      66667777      20160101000001VDEK, BERLIN
01 ";
```

Anschließend kann die Kernprüfung wie in 5.4.2 (Krips-Aufruf) oder 5.4.3 (Single-Aufruf) aufgeführt aufgerufen werden.

5.4.5 Aufrufparameter - XML-Kernprüfung

Bei Kernprüfungen von XML-Datenstrukturen ist als Vorlaufsatz nur das Verfahrensmerkmal zu übergeben. Den entsprechenden Wert hierfür können Sie der Tabelle entnehmen.

Verfahren	Verfahrensmerkmal
A1 Antrag Entsendung	A1A
A1 Ausnahmevereinbarung Arbeitgeber	A1A
A1 Ablehnung	A1S
A1 Bewilligung	A1S
A1 Antrag Beamte	A1A
A1 GME ein Arbeitgeber	A1A
A1 Antrag Flug und Kabinenbesatzung	A1A
A1 Antrag Seeleute	A1A
A1 Antrag Selbständige	A1A
A1 Antrag GME Global	A1A
A1 Antrag Grenzgänger	A1A
A1 Ausnahmevereinbarung Erwerbstätige und Rentner	A1A
DXRR	ARV
DXEB	ARV
DXAR	RVA
SVA Ablehnung	**SVS**
SVA Antrag Ausnahmevereinbarung	SVA
SVA Antrag Entsendung	SVA
SVA Antrag Entsendung Selbständige	SVA
SVA Ausnahmevereinbarung Selbständige	SVA
SVA Genehmigung	SVS

Als Datensatz muss die Fachstruktur übergeben werden. Zu beachten ist hier, dass die Namensräume definiert sein müssen, wenn man die Fachstruktur aus ihrem Rahmen löst.

Aus einem gesamten Antrag Entsendung muss also ein einzelner A1 Antrag (rot umrandet) gelöst werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--XML-Beispieldatei von XMLSpy generiert v2012 sp1 (http://www.altova.com)-->
<AGTOSV xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:A1="http://www.gkv-
datenaustausch.de/XMLSchema/Steuerungssatz_A1" xmlns:ant="http://www.gkv-
datenaustausch.de/XMLSchema/A1_Antrag/1.0" xmlns:btct="http://www.gkv-
datenaustausch.de/Grundsaetze_Technik/XML/Basistypen_Complex" xmlns="http://www.gkv-
datenaustausch.de/XMLSchema/SV_Header_AGTOSV/1.0" xsi:schemaLocation="http://www.gkv-
datenaustausch.de/XMLSchema/SV_Header_AGTOSV/1.0 SV_Header_AGTOSV.xsd"
Versionsnummer="aaaaa">
  - <SendungHeaderAGTOSV Versionsnummer="aaaaa">
      + <Vorlaufsatz>
      + <Kommunikationsdaten>
    </SendungHeaderAGTOSV>
  - <Sendungs_Body>
      <A1:Antraege>
        + <ant:A1_Antrag_Entsendung Versionsnummer="aaaaa">
      </A1:Antraege>
    </Sendungs_Body>
</AGTOSV>
```

Zudem müssen alle Namensräume, die für den gewünschten Antrag relevant sind, (gelb unterstrichen) aus dem Header der XML-Datei in die Fachstruktur übertragen werden.

String datensatz = "<ant:A1_Antrag_Entsendung Versionsnummer="1.1.0"
xmlns:ant="http://www.gkv-datenaustausch.de/XMLSchema/A1_Antrag/1.1"
xmlns:btct="http://www.gkv-datenaustausch.de/Grundsaetze_Technik/XML/Basistypen_Complex">
… </ant: A1_Antrag_Entsendung>";
String vorlaufsatz = "A1A";

Anschließend kann die Kernprüfung wie in 5.4.2 (Krips-Aufruf) oder 5.4.3 (Single-Aufruf) aufgeführt aufgerufen werden.

6. Logging

Die Kernprüfung verwendet die Logging API von Java (java.util.logging) als Loggingframework. Standardmäßig sind die Logausgaben deaktiviert. Die Logausgaben können aktiviert werden, in dem die Logkonfiguration in einer „logging.properties“-Datei dem Klassenpfad hinzugefügt wird. Die Klassen befinden sich im Package „de.deutscherv“, welches in der Loggingkonfiguration verwendet werden kann.

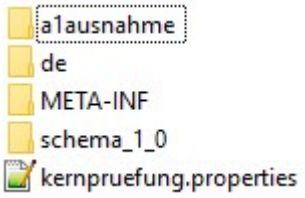
7. Schemavalidierung bei XML-Datenstruktur-Kernprüfungen

Bei der XML-Kernprüfung, werden standardmäßig die Nutzdaten (Datensätze) validiert. Die Kernprüfung wurde dahingehend konzipiert, dass davon ausgegangen wird, dass nach der Validierung der Datensatz syntaktisch korrekt ist. Bei einem Fehler im Aufbau des Datensatzes wird durch die Schemavalidierung der ReturnCode 6 zurückgegeben.

Ist der Datensatz nicht schemakonform, kann es dazu kommen, dass die Kernprüfung nicht korrekt funktioniert. Sind beispielsweise die Namespaces nicht korrekt gesetzt, können die einzelnen Felder nicht mehr gefunden werden. In der Folge wird keine Prüfung ausgeführt und der Datensatz wird als fehlerfrei zurückgegeben, obwohl das ggf. falsch ist.

Aus diesem Grund wird sehr dazu geraten die XML-Schemavalidierung aktiviert zu lassen.

Sollten dennoch die Deaktivierung der XML-Schemavalidierung gewünscht sein, kann das über die Datei „kernpruefung.properties“ konfiguriert werden. Die neu anzulegende Datei wird entweder auf Wurzelebene des Kernprüfung-JAR-Archivs (im Beispiel innerhalb der A1Ausnahme-Kernprüfung) oder im Klassenpfad gesucht.



In der Datei kann die Option „disable.xml.validation“ gesetzt werden. Wird hierfür „true“ vergeben wird die Nutzdatenvalidierung deaktiviert.

8. Versionierung

Die Versionsnummer der Kernprüfung setzt sich aus den folgenden Bestandteilen zusammen:

Hauptversionsnummer.Nebenversionsnummer.Revisionsnummer.Hotfixnummer

Beispiel: **1.4.12.1**

Hauptversionsnummer: Diese Ziffer wird bei grundlegenden technischen Änderungen an der Kernprüfung hochgezählt, beispielsweise, wenn sich die Aufrufschnittstelle des Programms grundlegend ändert. Diese Änderungen kommen selten vor.

Nebenversionsnummer: Die zweite Ziffer wird zu den jeweiligen Releaseterminen, wie 01.01.2018 und 01.07.2018, erhöht.

Revisionsnummer: Die dritte Ziffer wird bei kleineren Änderungen fachlicher und technischer Natur sowie Bugfixes inkrementiert.

Hotfixnummer: Die vierte Ziffer wird bei Kleinständerungen technischer Natur am Projekt (z.B. technische Fehlerprüfungen, Anpassungen an Build-Deskriptoren) inkrementiert. Diese Hotfixes gehen ohne das Durchlaufen der Testphasen und -stages in das Release ein.

9. Haftungsausschluss

Die Deutsche Rentenversicherung räumt dem Nutzer ein nicht ausschließliches und nicht übertragbares Recht zur Nutzung der hier angebotenen Programme ein. Der Nutzer darf genauso viele Kopien der Programme erstellen, wie er für die Nutzung in seinem Unternehmen benötigt. Die Nutzung ist unbefristet und kostenfrei. Die Ablauffähigkeit der Software ist unter Windows 7 im Allgemeinen gewährleistet, wird jedoch unter speziellen Konfigurationen nicht garantiert. Mängel an der Software wird die Deutsche Rentenversicherung Bund nach eigenem Ermessen beseitigen (Mängelmeldungen sind an kernpruefungen@dsrv.info zu richten). Die Haftung der Deutschen Rentenversicherung Bund wird ausschließlich auf Vorsatz und grobe Fahrlässigkeit begrenzt.

Die Deutsche Rentenversicherung Bund übernimmt insbesondere auch dann keine Haftung, wenn die Validierung der Struktur von Nutzdaten beim Anwender abgeschaltet wird.

10. Anhang: Programmdokumentation

Alle diese Kernprüfungen bestehen aus den im Folgenden beschriebenen Teilen, die schon fertig konfiguriert mit ausgeliefert werden.

10.1 Aufbau der einzelnen Kernprüfungen

10.1.1 Datenstruktur-Kernprüfungen Implementierung

Dabei handelt es sich um eine Java-Klasse, welche zur Ausführung einer Kernprüfung für einen Datensatz instanziiert und aufgerufen wird. Um hier einerseits eine gemeinsame Aufrufschnittstelle und andererseits eine einfache Implementierung zu gewährleisten, existieren hier je nach Art der Kernprüfung bereits folgende abstrakte Basisklassen:

- Für XML-basierte Kernprüfungen: AbstractKernpruefungXml
- Für Kernprüfungen von Datensätzen mit fixen Längen: AbstractKernpruefungFixLength

In dieser Java-Klasse für die Implementierung der Kernprüfung müssen im Wesentlichen nur die Pfade für folgende Dateien bereitgestellt werden:

- Konfiguration der Struktur für die zu prüfenden Datensätze (siehe Abschnitt 10.1.2)
- XML-basierte Konfigurationsdatei für die auszuführenden Prüfungen (siehe Abschnitt 10.1.3)
- Fehlertexte (siehe Abschnitt 10.1.4)

Optional können hier auch noch weitere Validatoren eingebunden werden. Konkrete Validatoren der technischen Kernprüfungs-Basis sind beispielsweise der XmlSchemaValidator und TotalLengthValidator. Derartige Validatoren prüfen allgemeine Kriterien des zu prüfendenden Datensatzes und werden noch vor der eigentlichen Prüfung ausgeführt.

Die Validierung der Struktur von Nutzdaten wird strengstens empfohlen. Derzeit ist die Validierung der Strukturen in Kernprüfungen, welche von der DRV zur Verfügung gestellt werden, nicht optional und somit nicht abschaltbar. Zur Validierung der Struktur wird in XML-Kernprüfungen zu diesem Zweck immer der XmlSchemaValidator eingesetzt. Die Funktion von XML-Kernprüfungen ist lediglich bei aktiver Schema Validierung der Nutzdaten gewährleitstet.

10.1.2 Konfiguration der Struktur

Die Konfiguration der Struktur für die zu prüfenden Datensätze erfolgt in Abhängigkeit von der Art der Kernprüfung wie folgt:

- Bei Kernprüfungen für Datensätze mit fixen Längen ist die Erstellung einer speziellen XML-Konfigurationsdatei erforderlich, in welcher der Aufbau des Datensatzes für eine weitere maschinelle Verarbeitung beschrieben ist. Diese XML-Konfigurationsdatei beschreibt somit die in den Datensätzen enthaltenen Bausteine bzw. Felder in Reihenfolge, Anzahl und Länge. Der Aufbau der XML-Konfigurationsdatei ist dabei in einer der technischen Basis der Kernprüfung beigelegten XML-Schemadatei DatensatzSchema.xsd festgelegt.
- XML-basierte Datensätze sind durch Ihren Aufbau selbstbeschreibend. Die weitere Prüfung gegen die XML-Schemadatei ist optional und wird mit dem im vorherigen Absatz beschriebenen XmlSchemaValidator realisiert.

10.1.3 Konfiguration der auszuführenden Prüfungen

Die Konfiguration der von einer Kernprüfung auszuführenden Prüfungen ist sowohl für XML-basierte als auch für Datensätze mit fixen Längen identisch und erfolgt stets in Form einer entsprechenden XML-Konfigurationsdatei.

Da innerhalb einer Kernprüfung typischerweise sehr viele Einzelprüfungen erfolgen, ist die Konfiguration der auszuführenden Prüfungen in der Regel die Hauptaufgabe bei der Erstellung einer Datenstruktur-Kernprüfungen. Hierbei sind folgende Dinge zu beachten:

- Der Aufbau der XML-Konfigurationsdatei ist in einer der technischen Basis der Kernprüfung beigelegten XML-Schemadatei pruefungen.xsd festgelegt.
- Zur Strukturierung der in der Regel zahlreichen Prüfungen ist es sinnvoll, sämtliche Prüfungen für einzelne Felder in jeweils einer eigenen Konfigurationsdatei auszulagern und diese Konfigurationsdateien mittels dem XML Mechanismus include in einer zentralen Konfigurationsdatei für die Kernprüfung zusammenzuführen
- Auszug aus einer Konfiguration

```
<?xml version="1.0"?>
<Feldpruefung pfad="/A1_Ablehnung/Steuerungsdaten/Datensatz_ID"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="pruefungen.definition.kernpruefung"
  xsi:schemaLocation="pruefungen.definition.kernpruefung ../../schema_1_0/pruefungen-fachlich.xsd">
  <Pruefungsdefinition id="Datensatz_ID_Fachliche_Pruefung">
    <Pruefungseinheit scope="feldbezogen" id="Datensatz_ID_Einheit01">
      <PruefungBestehtAus>
        <wert feldRef="."/>
        <erlaubteZeichen>
          <GrossbuchstabenABisZ href="technischepruefungen/wertelisten/GrossbuchstabenABisZ.xml"/>
          <KleinbuchstabenABisZ href="technischepruefungen/wertelisten/KleinbuchstabenABisZ.xml"/>
          <Ziffern href="technischepruefungen/wertelisten/Ziffern.xml"/>
          <Wert> </Wert>
          <Wert>.</Wert>
          <Wert>-</Wert>
          <Wert>/</Wert>
          <Wert>\</Wert>
        </erlaubteZeichen>
      </PruefungBestehtAus>
    </Pruefungseinheit>
  </Pruefungsdefinition>

  <Pruefungsfehlernummern>
    <Fehlernummer pruefungseinheitRef="Datensatz_ID_Einheit01" nummer="DXAA020" />
  </Pruefungsfehlernummern>
</Feldpruefung>
```

10.1.4 Fehlertexte

Zur besseren Pflegbarkeit werden die Fehlertexte nicht innerhalb der einzelnen Konfigurationsdateien für die Prüfungen verwaltet, sondern in einer eigenen Properties-Datei abgelegt. Diese Properties-Datei ist dabei so aufgebaut, dass die Fehlernummer der Zugriffsschlüssel (Key) für den jeweiligen Fehlertext ist.

```
DXAA020=DXAA020 Datensatz_ID enthält unzulässige Zeichen
DXAA030=DXAA030 Es ist nur der Wert "J" zulässig
```

10.1.5 Allgemeine Informationen

Alle Konfigurationsdateien (Fehlertexte, Struktur und Prüfungskonfiguration) sind in den Datenstruktur-Kernprüfungen entsprechend konfiguriert enthalten.

10.2 Kernprüfung Basis

Alle Datenstruktur-Kernprüfungen verwenden eine gemeinsame Basis. Deren Implementierung ist aufgegliedert in technische und fachliche Basis sowie weitere Komponenten. Die zugehörigen Module befinden sich im kernpruefung-basis-technisch-x.x.x.jar und sind verpflichtend im Klassenpfad bereitzustellen.

10.2.1 kernpruefung-interface

„kernpruefung-interface“ ist eine kleine Bibliothek, die die Schnittstelle der Kernprüfung beinhaltet. Jede Kernprüfung implementiert das Interface kernpruefung.Kernpruefung. Somit wird sichergestellt, dass sich für den Aufrufer keine Änderungen ergeben, selbst wenn sich die Prüfroutinen oder die Ablauflogik ändern. Bei neuen Versionen müssen lediglich die hinterlegten Bibliotheken ausgetauscht werden.

10.2.2 kernpruefung-basis-technisch

In „kernpruefung-basis-technisch“ sind das Modell der Kernprüfung sowie die groben Strukturen und die Ablauflogik abgelegt. Außerdem befinden sich hier auch die technischen Basisprüfungen, die für die einzelnen Feldprüfungen verwendet werden können. Es wurde nach dem Motto so viel wie nötig, so wenig wie möglich vorgegangen. Aktuell bietet die technische Basis rund 50 Basisprüfungen an, die über die Konfiguration mit den Fehlernummern verknüpft werden. Es wurde darauf geachtet, dass möglichst jeder Fall abgedeckt werden kann und die Anzahl der Basisprüfungen überschaubar bleibt. Zu den Basisprüfungen gehören sehr einfache Prüfungen wie ein Vergleich zweier Werte oder die zulässigen Zeichen eines Feldes, aber auch komplexe Datums- und Prüfzifferberechnungen (u.a. Versicherungsnummer, Absendernummer). Alle Prüfungen wurden mittels JavaDoc ausführlich dokumentiert.

10.2.3 kernpruefung-basis-fachlich

In „kernpruefung-basis-fachlich“ sind Prüfungsroutinen für Felder und Bausteine definiert. Beispielsweise werden Prüfungen für Vor- und Nachnamen, sowie für Postleitzahl oder Straßen angeboten. Kandidaten für die fachliche Basis wurden mittels einer aufwendigen Analyse ermittelt, bei der zwölf Datensätze mit über 900 Feldern miteinander verglichen wurden. Dieses Modul garantiert, dass gleiche Felder exakt gleich geprüft werden und deren Validierung nicht mehrfach implementiert werden muss.

10.2.4 datenaufbereitung

Die „datenaufbereitung“ ist eine eigens implementierte Bibliothek, die auch außerhalb der Kernprüfung zum Einsatz kommt. Hierbei wird ein Datensatz mit fester Länger über eine XML-Konfiguration eingelesen und aufbereitet ausgegeben. Die Kernprüfung macht sich dies zu Nutze und verwendet die darin definierten Interfaces für den Parse-Algorithmus. Es wird das Jar „datensatzaufbereitung-api“ verwendet.

11. Anlage: Lizenztexte

11.1 Apache 2.0

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

11.2 MPL 2.0

Mozilla Public License Version 2.0

==========

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in

a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having

made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the

distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under

the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for

the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any

liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such

Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

****

* *

* 6. Disclaimer of Warranty *

```
* ------------------------                                    *
*                                                             *
* Covered Software is provided under this License on an "as is"  *
* basis, without warranty of any kind, either expressed, implied, or  *
* statutory, including, without limitation, warranties that the  *
* Covered Software is free of defects, merchantable, fit for a  *
* particular purpose or non-infringing. The entire risk as to the  *
* quality and performance of the Covered Software is with You.  *
* Should any Covered Software prove defective in any respect, You  *
* (not any Contributor) assume the cost of any necessary servicing,  *
* repair, or correction. This disclaimer of warranty constitutes an  *
* essential part of this License. No use of any Covered Software is  *
* authorized under this License except under this disclaimer.  *
*                                                             *
***************************************************************************

***************************************************************************
*                                                             *
* 7. Limitation of Liability                                  *
* --------------------------                                  *
*                                                             *
* Under no circumstances and under no legal theory, whether tort  *
* (including negligence), contract, or otherwise, shall any  *
* Contributor, or anyone who distributes Covered Software as  *
* permitted above, be liable to You for any direct, indirect,  *
```

* special, incidental, or consequential damages of any character *
* including, without limitation, damages for lost profits, loss of *
* goodwill, work stoppage, computer failure or malfunction, or any *
* and all other commercial damages or losses, even if such party *
* shall have been informed of the possibility of such damages. This *
* limitation of liability shall not apply to liability for death or *
* personal injury resulting from such party's negligence to the *
* extent applicable law prohibits such limitation. Some *
* jurisdictions do not allow the exclusion or limitation of *
* incidental or consequential damages, so this exclusion and *
* limitation may not apply to You. *
* *
****

8. Litigation

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE

file in a relevant directory) where a recipient would be likely to look

for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as

defined by the Mozilla Public License, v. 2.0.

11.3 EDL 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

11.4 MIT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

11.5 BSD-2-Clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12. Anlage: Änderungshistorie

Version	Datum	Änderung
1.0.1	18.12.2017	• Einführung der Änderungshistorie • Änderung der Schriftart • Neues Kapitel 8 Versionierung • Aktualisierung der KP-Verzeichnisse unter Punkt 3.2
1.1.0	25.01.2018	• Kapitel 2: Anpassung der Grafik • Kapitel 3.3: Änderung der Strukturbeschreibung auf einzelnes SingleJar • Kapitel 8: Neues Kapitel Validierung • Formatierung
1.1.1		• Kapitel 4: Drittabhängigkeiten • Kapitel 7: Logging
1.1.2	15.05.2018	• Kapitel 4: Drittabhängigkeiten • Kapitel 5.4: Aufrufmöglichkeiten
1.2.0	06.07.2018	• Kapitel 3: Versionsnummern aktualisiert • Kapitel 4: Drittabhängigkeiten aktualisiert • Kapitel 5: Aufrufmöglichkeiten aktualisiert • Kapitel 6: Performance entfallen
1.2.1	09.08.2019	• Kapitel 4 Drittabhängigkeiten • Neue Anlage Lizenztexte APACHE 2.0 + MPL 2.0
2.0.0	19.06.2020	• Umstellung auf Java 8 • Kapitel 2: Anpassung der Grafik • Kapitel 3: Anpassung der Grafik • Kapitel 5: Systemvoraussetzungen • Kapitel 8: Versionierung 4. Stelle • Kapitel 10: Umbenennung der Module
2.1.0	13.11.2020	•
2.2.0	28.09.2023	• Kapitel 3: Regelwerk und einzelne Datenstruktur-Kernprüfungen

		• Kapitel 4: Drittabhängigkeiten • Kapitel 5: Systemvoraussetzungen • Kapitel 5: Aufrufparameter – XML-Kernprüfungen
2.3.0	18.11.2024	• Kapitel 3: Regelwerk und einzelne Datenstruktur-Kernprüfungen • Kapitel 4: Drittabhängigkeiten • Kapitel 5: Aufrufparameter XML-Kernprüfungen
2.4.0	07.07.2025	• Kapitel 3: Grafik der Komponenten • Kapitel 4: Drittabhängigkeiten • Kapitel 4.1: Neues Kapitel • Kapitel 5.1: Systemvoraussetzungen • Kapitel 11.3: Anlage: Lizenztexte – Neues Kapitel
2.5.0	03.08.2026	• Kapitel 3.2: Regelwerk und einzelne Datenstruktur-Kernprüfungen • Kapitel 4: Drittabhängigkeiten • Kapitel 5.4.5: Aufrufparameter - XML-Kernprüfung • Kapitel 11.4: Anlage: Lizenztexte – Neues Kapitel • Kapitel 11.5: Anlage: Lizenztexte – Neues Kapitel