



Institut für Qualitätssicherung und  
Transparenz im Gesundheitswesen

# **Technische Dokumentation zur Spezifikation der einrichtungsbezogenen QS-Dokumentation**

Erfassungsjahr 2017

Erstellt im Auftrag des  
Gemeinsamen Bundesausschusses

Stand: 30. Juni 2017

---

# Impressum

**Thema:**

Technische Dokumentation zur Spezifikation der einrichtungsbezogenen QS-Dokumentation.  
QS-Filter, QS-Dokumentation. Erfassungsjahr 2017

**Auftraggeber:**

Gemeinsamer Bundesausschuss

**Datum der Veröffentlichung:**

30. Juni 2017

**Herausgeber:**

IQTIG – Institut für Qualitätssicherung  
und Transparenz im Gesundheitswesen

Katharina-Heinroth-Ufer 1  
10787 Berlin

Telefon: (030) 58 58 26-0  
Telefax: (030) 58 58 26-999

[info@iqtig.org](mailto:info@iqtig.org)

<https://www.iqtig.org>

**Hinweis:**

Aus Gründen der leichteren Lesbarkeit wird im Folgenden auf eine geschlechtsspezifische Differenzierung verzichtet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für beide Geschlechter.

# Inhaltsverzeichnis

Tabellenverzeichnis .....	7
Abbildungsverzeichnis .....	9
Abkürzungsverzeichnis.....	10
Änderungsindex .....	12
Leseanleitung.....	16
1 Einleitung.....	18
1.1 Spezifikationsbegriff.....	18
1.1.1 Benennungsschema für Spezifikationspakete.....	19
1.1.2 Benennungsschema für Spezifikationskomponenten.....	20
1.2 Zielsetzung und Zielgruppe .....	21
1.3 Releaseplanung.....	21
A Prozesse.....	23
1 QS-Dokumentation.....	24
1.1 Auslösung.....	24
1.2 Erfassung.....	25
1.2.1 Gestaltung von Eingabemasken.....	25
1.2.2 Einrichtungsidentifizierende Daten.....	29
1.2.3 QS-Daten.....	30
1.2.4 Plausibilitätsprüfungen.....	30
1.3 Export der Daten.....	31
1.3.1 Erzeugen der Exportdatei.....	32
1.3.2 Datenprüfung.....	36
1.3.3 XML-Verschlüsselung.....	36
1.3.4 Ausgangsvalidierung.....	37
1.3.5 Beispiele für Exportdateien.....	38
1.4 Datenübermittlung.....	38
1.4.1 Dateibenennung.....	38
1.4.2 Datenversand via E-Mail .....	38
1.4.3 Datenversand via gesicherter Schnittstellen – Arztpraxen/MVZ für kollektivvertraglich erbrachte Leistungen .....	39

1.5	Rückprotokollierung.....	40
1.5.1	Funktion von Empfangsbestätigung und Datenflussprotokoll im Datenfluss.....	40
B	Komponenten.....	44
1	QS-Filter.....	48
1.1	Anmerkungen zur Struktur der Spezifikationsdatenbank für QS-Filter .....	48
1.2	Grundlegende Tabellen der Datenbank .....	48
1.2.1	Module (Datensätze der QS-Dokumentation).....	48
1.2.2	Struktur der Datensatzdefinitionen .....	50
1.2.3	Funktionen .....	55
1.2.4	OPS-Listen .....	56
1.2.5	ICD-Listen .....	56
1.2.6	FAB-Listen.....	57
1.2.7	EBM-Listen.....	58
1.2.8	Entgelt-Listen.....	58
1.2.9	Versionsverwaltung.....	58
1.2.10	Meta-Tabellen.....	59
1.2.11	DB-Änderungen gegenüber der Vorgängerversion.....	60
1.3	Der QS-Filter-Datensatz.....	60
1.3.1	Der QS-Filter-Eingangsdatsatz .....	60
1.4	Der Algorithmus zur Ermittlung der Dokumentationspflicht.....	66
1.4.1	Einleitung und Überblick.....	66
1.4.2	Verfahrensbezogene Einschlusskriterien.....	67
1.4.3	Administrative Einschlusskriterien.....	70
1.4.4	Struktur und Syntax der Auslösebedingungen .....	71
1.4.5	Fehlerprüfung.....	74
2	QS-Dokumentation.....	77
2.1	Anmerkungen zur Struktur der Spezifikation zur QS-Dokumentation.....	77
2.2	Datenfeldbeschreibung .....	77
2.2.1	Dokumentationsmodule (Datensätze) .....	79
2.2.2	Teildatensätze.....	81
2.2.3	Datenfelder (Bogenfelder) .....	82
2.2.4	Überschriften.....	89

2.2.5	Ausfüllhinweise.....	89
2.3	Plausibilitätsprüfungen.....	90
2.3.1	Die Regeltabelle.....	91
2.3.2	Regelsyntax.....	92
2.3.3	Funktionen.....	97
2.3.4	Syntaxvariablen.....	97
2.3.5	Einzelregeln.....	97
2.3.6	Teildatensatzübergreifende Regeln.....	98
2.3.7	Feldgruppenregeln.....	98
2.3.8	Prüfung von Feldeigenschaften.....	104
2.3.9	Verfahren für die Evaluation von Regeln.....	107
2.4	Exportfeldbeschreibung.....	109
2.4.1	Exportmodule.....	109
2.4.2	Modulauslöser, Dokumentations- und Exportmodule.....	110
2.4.3	Exportdatensatz.....	111
2.5	Versionierung.....	111
2.5.1	Grundlegende Definitionen.....	111
2.5.2	Delta-Informationen zur vorhergehenden Version.....	112
2.6	Administrative Objekte.....	113
2.6.1	CSV/XML-Mapping in der Spezifikationsdatenbank (QSDOK).....	113
2.6.2	Datenservices.....	114
3	XML-Schema.....	116
3.1	Kompositionsmodell.....	116
3.2	Schnittstellen.....	117
3.3	Darstellung der XML-Struktur.....	119
4	Tools.....	120
4.1	Java-Installation.....	120
4.2	Datenprüfprogramm.....	121
4.2.1	Umfang der Prüfungen.....	121
4.2.2	Ausgangskontrolle vor Versand.....	121
4.2.3	Programmaufruf.....	122
4.2.4	Verzeichnisstruktur.....	125

4.2.5	Ausgabe.....	126
4.2.6	Grafische Oberfläche .....	127
4.2.7	Programmierschnittstelle – API.....	127
4.3	Verschlüsselungspaket .....	127
4.3.1	XPacker – XML-Verschlüsselung .....	128
4.3.2	TPacker – Transportverschlüsselung.....	130
4.3.3	TPacker und XPacker ohne externe Abhängigkeiten .....	133
4.3.4	Programmierschnittstelle – API.....	133
4.3.5	GPager.....	133
4.4	LE-Pseudonymisierungsprogramm.....	134
C	Anhang.....	135
	Glossar.....	135

## Tabellenverzeichnis

Tabelle 1: Informationen aus der Datenbank, welche im GUI verwendet werden .....	25
Tabelle 2: Pseudonymisierungsverfahren des Verfahrens QS WI.....	33
Tabelle 3: XML-Schemata zur Prüfung vor der Verschlüsselung .....	36
Tabelle 4: Asymmetrische Verschlüsselung der XML-Elemente.....	37
Tabelle 5: XML-Schemata zur Prüfung nach der Verschlüsselung.....	37
Tabelle 6: Struktur der Tabelle Modul .....	49
Tabelle 7: Struktur der Tabelle Ds.....	51
Tabelle 8: Struktur der Tabelle Tds.....	51
Tabelle 9: Struktur der Tabelle BasisTyp.....	52
Tabelle 10: Struktur der Tabelle Schluessel.....	53
Tabelle 11: Struktur der Tabelle SchluesselWert.....	54
Tabelle 12: Struktur der Tabelle Version.....	59
Tabelle 13: Ausschnitt der Tabelle Ds.....	60
Tabelle 14: Felder des QS-Filter-Eingangsdatensatzes nach § 301 .....	61
Tabelle 15: Felder des QS-Filter-Eingangsdatensatzes nach § 295 .....	64
Tabelle 16: Struktur der Tabelle ModulAusloeser.....	68
Tabelle 17: Struktur der Tabelle AdminKriterium.....	71
Tabelle 18: Basistypen der Variablen .....	71
Tabelle 19: Präzedenz und Assoziativität der Operatoren.....	73
Tabelle 20: Fehlercodes des QS-Filters.....	74
Tabelle 21: Struktur der Tabelle Modul.....	79
Tabelle 22: Inhalte der Tabelle BogenTyp.....	81
Tabelle 23: Struktur der Tabelle BogenFeld.....	82
Tabelle 24: Struktur der Tabelle Feld.....	84
Tabelle 25: Struktur der Tabelle BasisTyp.....	86
Tabelle 26: Struktur der Tabelle Schluessel.....	86
Tabelle 27: Struktur der Tabelle SchluesselWert.....	88
Tabelle 28: Struktur der Tabelle Abschnitt.....	89
Tabelle 29: Arten von Hinweistypen.....	90
Tabelle 30: Tabelle RegelTyp.....	90
Tabelle 31: Struktur der Tabelle Regeln.....	91
Tabelle 32: Struktur der Tabelle RegelFelder .....	92
Tabelle 33: Basistypen der Datenfelder in den Plausibilitätsregeln.....	93
Tabelle 34: Präzedenz und Assoziativität der Operatoren.....	94
Tabelle 35: Typen von Feldgruppen.....	99
Tabelle 36: Struktur der Tabelle FeldGruppe.....	100
Tabelle 37: Struktur der Tabelle FeldgruppeFelder.....	100
Tabelle 38: Formale Definition einer Feldgruppe .....	102
Tabelle 39: Struktur der Tabelle ExportModul.....	109

Tabelle 40: Datenfelder zum Status des Leistungserbringers .....	110
Tabelle 41: Verwendbare Schemata und Ablageort.....	117
Tabelle 42: Weiche Schemavarianten für das Datenprüfprogramm.....	119

## Abbildungsverzeichnis

Abbildung 1: Serielles Datenflussmodell für die sektorenübergreifende Qualitätssicherung....	31
Abbildung 2: Beispiel einer Empfangsbestätigung.....	40
Abbildung 3: Tabellen und Relationen der Datenfeldbeschreibung.....	78
Abbildung 4: Algorithmus zur Evaluation von Plausibilitätsregeln .....	108
Abbildung 5: Dateiordner der Schnittstellen-Schemata.....	117
Abbildung 6: Weiche Schemavarianten für das DPP .....	125
Abbildung 7: Verzeichnisstruktur eines DPP-Ordners.....	126
Abbildung 8: Beispiel für eine Index.html Datei im Ordner <output_path>/html.....	127
Abbildung 9: Grafische Oberfläche des Datenprüfprogramms .....	127
Abbildung 10: Verschlüsselung eines XML-Elements (qs_data).....	129

## Abkürzungsverzeichnis

Abkürzung	Bedeutung
AG	Arbeitsgruppe
AG Qesü-RL	Arbeitsgruppe Qesü-RL des G-BA
BAS	Bundesauswertungsstelle
BE	Bundesebene
BSNR	Betriebsstättennummer
DAS	Datenannahmestelle
DAS-SV	Datenannahmestellen für selektivvertraglich erbrachte Leistungen
DIMDI	Deutsches Institut für Medizinische Dokumentation und Information
DRG	Diagnosis Related Groups (diagnosebezogene Fallgruppen)
EBM	Einheitlicher Bewertungsmaßstab
FU	Follow-up
G-BA	Gemeinsamer Bundesausschuss
GKV	Gesetzliche Krankenversicherung
GOP	Gebührenordnungsposition
GUID	Globally Unique Identifier
ICD	International Classification of Diseases (Internationale Klassifikation der Krankheiten und verwandter Gesundheitsprobleme)
ICD-10-GM	Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme – 10. Revision – German Modification
ID	Identifikationsnummer
IK	Institutionskennzeichen
IKNR	Institutionskennzeichen
iv	Indirekte Verfahren
KBV	Kassenärztliche Bundesvereinigung
KH	Krankenhaus
KHEntgG	Gesetz über die Entgelte für voll- und teilstationäre Krankenhausleistungen (Krankenhausentgeltgesetz)
KIS	Krankenhausinformationssystem

Abkürzung	Bedeutung
KV	Kassenärztliche Vereinigung
KVDT	Kassenärztliche Vereinigung-Datentransfer (Datenformat)
LANR	Lebenslange Arztnummer
LE	Leistungserbringer
LKG	Landeskrankenhausgesellschaft(en)
LQS	Landesgeschäftsstelle für Qualitätssicherung
MVZ	Medizinisches Versorgungszentrum
OPS	Operationen- und Prozedurenschlüssel
OR	ODER-Operator
PB	Probetrieb
PID	Patientenidentifizierende Daten
PR	Arztpraxis
Qesü-RL	Richtlinie (des GBA) gemäß § 92 Abs. 1 Satz 2 Nr. 13 i.V.m. § 137 Abs. 1 Nr. 1 SGB V über die einrichtungs- und sektorenübergreifenden Maßnahmen der Qualitäts-sicherung
QS	Qualitätssicherung
QS WI	QS-Verfahren <i>Vermeidung nosokomialer Infektionen – postoperative Wundinfektionen</i>
RAM	Random Access Memory (Arbeitsspeicher)
RL	Richtlinie
RSA	Verfahren zur Datenverschlüsselung, entwickelt von R. Rivest, A. Shamir und L. Adleman
SGB	Sozialgesetzbuch
SGB V	Sozialgesetzbuch Fünftes Buch
SQG	Sektorenübergreifende Qualität im Gesundheitswesen
SWA	Softwareanbieter
TB	Testbetrieb
TDS	Teildatensatz
VST	Vertrauensstelle
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation (Programmiersprache zur Transformation von XML-Dokumenten)

# Änderungsindex

Änderungen der Datenbanken im Vergleich zur Vorversion 2017 V01 Beta lassen sich anhand der Delta-Tabellen nachvollziehen.

## Neue Abschnitte in der Version 2017 V01:

- Integration eines neuen Abschnitts „Gestaltung von Eingabemasken“ (Abschnitt 1.2.1)
- Ergänzung eines neuen Abschnittes zur Rückprotokollierung (Abschnitt 1.5.1)
- Ergänzung neuer Abschnitte zu administrativen Objekten in der Datenbank zur QS-Dokumentation (Abschnitt 2.6.1 CSV/XML-Mapping in der Spezifikationsdatenbank (QSDOK) und 2.6.2 Datenservices)
- Ergänzung neuer Abschnitte zu Tools (Abschnitt 4.1 Java-Installation, 4.2 Datenprüfprogramm, 4.3 Verschlüsselungspaket und 4.4 LE-Pseudonymisierungsprogramm)
- Anpassungen an die technische Dokumentation zur Basisspezifikation für Leistungserbringer, z.B. erlaubte Zeichen des ASCII-Formats mit einem Kode  $\geq 32$  in Zeichenketten (Basistyp TEXT)

## Kapitelübergreifende Änderungen in der Version 2017 V01 Beta:

- Der Begriff „Leistungsbereich“ wird durch den Begriff „QS-Verfahren“ ersetzt. Der Begriff QS-Filter-Leistungsbereich wird durch den Begriff QS-Filter-Modulauslöser ersetzt.
- Anpassung von Beispielen, Einfügen von Verweisen, Redaktionelle Änderungen

Konkrete Informationen zu den inhaltlichen Änderungen sind der Spezifikationskomponente Uebersicht\_Aenderungen zu entnehmen.

Die spezifischen Änderungen der vorliegenden technischen Dokumentation werden im Folgenden mit Bezug zur jeweiligen Version dargestellt.

Änderung	Kapitel/Abschnitt	Version
Beschreibung der einrichtungsbezogenen QS-Dokumentation	-	2017 V01 Alpha
Ergänzung der Einleitung um Informationen zum Verfahren QS WI und den hierbei zu berücksichtigenden Erhebungsinstrumenten	1 Einleitung	2017 V01 Beta
Ergänzen von Meilensteinen und Fristen im Rahmen der Releaseplanung	1.3 Releaseplanung	2017 V01 Beta
Einfügen eines neuen Abschnitts „Einrichtungsidifizierende Daten“	A 1.2.2	2017 V01 Beta
Integration eines neuen Kapitels „Gestaltung von Eingabemasken“	A 1.2.1	<b>2017 V01</b>
Einfügen eines neuen Abschnitts „QS-Daten“	A 1.2.3	2017 V01 Beta

Änderung	Kapitel/Abschnitt	Version
Klarstellung zur Dokumentation belegärztlicher Fälle; diese werden in diesem Verfahren z.B. über eine webbasierte Lösung erfasst. Diese wird durch die KVen zur Verfügung gestellt.	A 1.2.2	<b>2017 V01</b>
Ergänzen eines Hinweises zum seriellen Datenflussmodell der Qesü-RL, da der Datenfluss über die VST entfällt	A 1.3	2017 V01 Beta
Ergänzung des Absatzes zu Sonderzeichen in XML	A 1.3.1	<b>2017 V01</b>
Anpassung einer Fußnote an die aktuelle Qesü-RL	A 1.3.2	<b>2017 V01</b>
Einfügen neuer Unterabschnitte zu Abschnitt A 1.3 „Export der Daten“ (Erzeugen der Exportdatei, Datenprüfung, XML-Verschlüsselung, Ausgangsvalidierung, Beispiele für Exportdateien)	A 1.3.1, A 1.3.2, A 1.3.3, A 1.3.4, A 1.3.5	2017 V01 Beta
Einfügen eines neuen Abschnitts A 1.4 „Datenübermittlung“ inkl. Unterabschnitte (Dateibenennung, Datenversand via E-Mail, Datenversand via gesicherter Schnittstellen – Arztpraxen/MVZ für kollektivvertraglich erbrachte Leistungen)	A 1.4, A 1.4.1, A 1.4.2, A 1.4.3	2017 V01 Beta
Einfügen eines neuen Abschnitts „Rückprotokollierung“	A 1.5	2017 V01 Beta
Beschreibung der Struktur der Tabelle Modul (QSF-Datenbank)	B 1.2.1	2017 V01 Beta
Anpassung einer Fußnote an die aktuelle Qesü-RL	B 1.2.1	<b>2017 V01</b>
Ergänzung der Beschreibung der Tabelle Tds um die Attribute fkEindeutigTdsFeld und sortierNr	B 1.2.2	2017 V01 Beta
Klarstellung der Referenzierung verschiedener externer Quellen des Schlüssels bzw. Datenfeldes PERSONENKREIS	B 1.2.2	<b>2017 V01</b>
Anpassung des Abschnitts „OPS-Listen“	B 1.2.3	2017 V01 Beta
Einfügen eines neuen Abschnitts zu Syntaxfunktionen in der QSF	B 1.2.3	2017 V01 Beta
Anpassung des Abschnitts „ICD-Listen“	B 1.2.5	2017 V01 Beta
Ergänzung eines neuen Abschnitts „FAB-Listen“	B 1.2.6	2017 V01 Beta

Änderung	Kapitel/Abschnitt	Version
Ergänzung des Hinweises zu „Datenverlust durch Fehlbezug und falsche Falleinschränkung“ um eine Information zur Teilbedingung zur Identifikation gesetzlich versicherter Patienten	B 1.3	2017 V01 Beta
Anpassung des Abschnitts zum „QS-Filter-Eingangsdatensatz nach §301 Abs. 3 SGB V“ an die QS-Filterdatenbank zur einrichtungsbezogenen QS-Dokumentation (z.B. Löschen nicht relevanter Felder und Teildatensätze)	B 1.3.1	2017 V01 Beta
Anpassung des Abschnitts zum „QS-Filter-Eingangsdatensatz nach §295 SGB V“ an die QS-Filterdatenbank zur einrichtungsbezogenen QS-Dokumentation (z.B. Löschen nicht relevanter Felder und Teildatensätze)	B 1.3.1	2017 V01 Beta
Ergänzung des Abschnitts zum „QS-Filter-Eingangsdatensatz nach §295 SGB V“ um Referenzen auf die Datensatzbeschreibung KVTD am Beispiel der Version 5.15 vom 07.02.2017	B 1.3.1	2017 V01 Beta
Integration der beiden TDS-Felder PERSONENKREIS und VERSICHERTENIDNEU in den Eingangsdatensatz nach § 301 und in den Eingangsdatensatz § 295	B 1.3.1	<b>2017 V01</b>
Ergänzung des Abschnitts zum „QS-Filter-Eingangsdatensatz nach §295 SGB V“ um einen neuen Abschnitt sowie um ein Beispiel zu Primär- und Sekundär Diagnosen	B 1.3.1, S. 66	2017 V01 Beta
Ergänzung des Hinweises zu Einschlusskriterien mit KASSEIKNR um die Angaben zur VERSICHERTENART und zum PERSONENKREIS	B 1.4.2	2017 V01 Beta
Löschen des Abschnitts „Ausnahmen für einzelne QS-Filter-Leistungsbereiche“	B 1.4.2	2017 V01 Beta
Ergänzung des Abschnitts „Identifizierung der Fallart und der Art der Leistungserbringung“ um identifizierende Merkmale des Datensatzes DATENSATZ_295	B 1.4.2	2017 V01 Beta
Einfügen eines Hinweises „Hinweis zu DATENSATZ_295“ zur Differenzierung zwischen ambulanten und stationären	B 1.4.2	2017 V01 Beta

Änderung	Kapitel/Abschnitt	Version
Einfügen eines Hinweises „Hinweis zu DATENSATZ_295“ zum Attribut <code>kollektiv</code>	B 1.4.2	2017 V01 Beta
Anpassung des Hinweises zu Einschlusskriterien mit <code>KASSEIKNR</code> , <code>VERSICHERTENIDNEU</code> und <code>PERSONENKREIS</code>	B 1.4.2	<b>2017 V01</b>
Anpassung der Erläuterung des Attributs <code>Modulausloeser.fkDs</code>	B 1.4.2, Tabelle 16	2017 V01 Beta
Ergänzung des Abschnitts „Diagnose-, Prozeduren- und Abrechnungslisten“ um Fachabteilungslisten	B 1.4.4	2017 V01 Beta
Ergänzung der BasisTypen <code>JAHRDATUM</code> und <code>QUARTDATUM</code>	B 1.4.4, Tabelle 18	2017 V01 Beta
Anpassung der Fehlerprüfung in Bezug auf das korrekte Erfassungsjahr der Spezifikation	B 1.4.5, Tabelle 20	2017 V01 Beta
Anpassung des Benennungsschemas für die Datenbank zur QS-Dokumentation	B 2.1	<b>2017 V01</b>
Anpassung einer Fußnote an die aktuelle Qesü-RL	B 2.2	<b>2017 V01</b>
Anpassung des Abschnitts zu Syntaxfunktionen in der QSDOK	B 2.3.3	2017 V01 Beta
Konkretisierung der Fehlerprüfungen in Bezug auf Bogen-, Ersatz- oder Zusatzfelder	B 2.3.8	<b>2017 V01</b>
Einfügen eines neuen Abschnitts zur Differenzierung zwischen Modulauslöser, Dokumentationsmodul und Exportmodul	B 2.4.2	2017 V01 Beta
Anpassung des Hinweises zur Vorbelegung der Datenfelder zum Status des Leistungserbringers	B 2.4.2	<b>2017 V01</b>
Integration neuer Abschnitte zu administrativen Objekten in der Datenbank zur QS-Dokumentation	B 2.6	<b>2017 V01</b>
Überarbeitung des Kapitels „XML-Schema“	B 3	<b>2017 V01</b>
Überarbeitung des Kapitels „Tools“	B 4	<b>2017 V01</b>
Beschreibung von möglichen Spezifikationskomponenten eines Spezifikationspaketes	B Komponenten	2017 V01 Beta
Erläuterung von Abkürzungen/Begriffen	Glossar; Abkürzungsverzeichnis	2017 V01 Beta

# Leseanleitung

Diese technische Dokumentation orientiert sich in ihrem Aufbau an den Abläufen der Erfassung und Übermittlung der Qualitätssicherungsdaten. Ziel dieser Struktur ist es, eine nachvollziehbare und logische Sicht auf die Umsetzung und Durchführung der beschriebenen Schritte zu gewährleisten. Die Prozesse und Unterprozesse werden im Abschnitt A Prozesse beschrieben und spiegeln die reale, chronologische Abfolge wider. Jede Prozessbeschreibung berücksichtigt zudem die unterschiedlichen Komponenten, die für die Umsetzung benötigt und in Abschnitt B Komponenten detailliert beschrieben werden. Im Abschnitt C Anhang wird ein Glossar mit den wichtigen Begriffen zum Themenbereich der Spezifikation zur Verfügung gestellt.

Für eine korrekte Umsetzung der Spezifikation ist es notwendig, die Dokumentation entsprechend ihrer Anordnung von Prozessen zu Komponenten zu befolgen. Einige Bereiche, die sich ausschließlich an bestimmte Zielgruppen richten, sind entsprechend gekennzeichnet.

## Legende

Die in dieser Dokumentation verwendeten Symbole heben bestimmte Aspekte bei der Umsetzung der Spezifikation hervor.



### Achtung

Beschreibt Ursache, Folge und Vermeidung einer besonderen Fehlanwendung, die zu Problemen bei der Implementierung oder Ähnlichem führen kann.

---



### Hinweis

Nützliche Informationen, Tipps oder Ratschläge zur Anwendung. Keine wesentlichen oder für das korrekte Funktionieren erforderlichen Informationen.

---

### Beispiel:

Beispiele sind ein Hilfsmittel, um zuvor vermittelte Informationen oder konkrete Abschnitte der Anwendung zu verdeutlichen.

---

---

<b>LE</b> →	<b>Zielgruppe LE</b> Markiert nur für folgende Zielgruppe relevanten Bereich: Leistungserbringer (LE) im Allgemeinen
-------------	--

---

<b>VÄ</b> →	<b>Zielgruppe VÄ</b> Markiert nur für folgende Zielgruppe relevanten Bereich: Vertragsärztin / Vertragsarzt (VÄ)
-------------	--

---

<b>KH</b> →	<b>Zielgruppe KH</b> Markiert nur für folgende Zielgruppe relevanten Bereich: Krankenhaus (KH)
-------------	--

---

<b>DAS</b> →	<b>Zielgruppe DAS</b> Markiert nur für folgende Zielgruppe relevanten Bereich: Datenannahmestelle (DAS) im Allgemeinen
--------------	--

---

<b>KV</b> →	<b>Zielgruppe KV</b> Markiert nur für folgende Zielgruppe relevanten Bereich: Kassenärztliche Vereinigung (KV)
-------------	--

---

<b>LQS LKG</b> →	<b>Zielgruppe LQS/LKG</b> Markiert nur für folgende Zielgruppe relevanten Bereich: Landesgeschäftsstelle für Qualitätssicherung (LQS) Landeskrankenhausgesellschaften (LKG)
----------------------	--

---

# 1 Einleitung

Die vorliegende technische Dokumentation beschreibt die Spezifikation zur einrichtungs- und sektorenübergreifenden Qualitätssicherung gemäß §§ 136 ff. SGB V.

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation umfasst alle Komponenten im Zusammenhang mit der Datenerfassung, d.h. von der Bestimmung einer Dokumentationspflicht bis hin zur Rückprotokollierung übermittelter Datensätze. Die vorliegende technische Dokumentation beschreibt die hierbei relevanten Prozesse und Komponenten.

Regelungsbereich der Spezifikation der einrichtungsbezogenen QS-Dokumentation ist das Verfahren *Vermeidung nosokomialer Infektionen – postoperative Wundinfektionen* gemäß der Richtlinie zur einrichtungs- und sektorenübergreifenden Qualitätssicherung (Qesü-RL)<sup>1</sup>, die vom Gemeinsamen Bundesausschuss (G-BA) beschlossen wird. Im Erfassungsjahr 2017 beinhaltet die Spezifikation der einrichtungsbezogenen QS-Dokumentation daher Module des Verfahrens *Vermeidung nosokomialer Infektionen – postoperative Wundinfektionen* (QS WI).

Den Auftrag zu dem neuen Qualitätssicherungsverfahren QS WI gab der G-BA im Dezember 2015. Der Regelbetrieb startete am 01.01.2017.

Das QS-Verfahren wird über verschiedene Erhebungsinstrumente abgebildet:

- die fallbezogene Dokumentation beim Leistungserbringer
- die Erhebung von Sozialdaten bei den Krankenkassen
- die jährliche einrichtungsbezogene QS-Dokumentation

Ziel des QS-Verfahrens ist es, die Qualität der Maßnahmen ambulanter und stationärer Leistungserbringer zur Vermeidung nosokomialer Infektionen, insbesondere postoperativer Wundinfektionen zu messen, vergleichend darzustellen und zu bewerten. Dafür werden bundesweit Falldaten von Patientinnen und Patienten zusammengeführt, bei denen nach ambulanter oder stationärer Behandlung eine solche Wundinfektion aufgetreten ist, und mit der Ausgangsbehandlung in Bezug gesetzt.

Die vorliegende Spezifikation bildet das Erhebungsinstrument der jährlichen einrichtungsbezogenen QS-Dokumentation beim Leistungserbringer ab. Für die Nutzung von Sozialdaten bei den Krankenkassen gemäß § 299 SGB V sowie die fallbezogene QS-Dokumentation beim Leistungserbringer werden unabhängige Spezifikationen veröffentlicht.<sup>2</sup> Für eine spezifikationskonforme Umsetzung von QS-Softwareprodukten sind alle Spezifikationskomponenten zu berücksichtigen.

## 1.1 Spezifikationsbegriff

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation ist die Gesamtheit aller Vorgaben, nach denen die Bestimmung der dokumentationspflichtigen Einrichtungen, die QS-Doku-

---

<sup>1</sup> <https://www.g-ba.de/informationen/richtlinien/72/>

<sup>2</sup> <https://iqtig.org/datenservice>

mentation selbst und die Übermittlung der Daten erfolgen sollen, bezogen auf ein Erfassungsjahr. Die Dokumentationspflicht einer Einrichtung wird ausgelöst, wenn mindestens ein Fall den definierten Auslösekriterien entspricht. Die Zuordnung eines Falles zu einer Spezifikation richtet sich nach dem Kalenderjahr der Aufnahme in das Krankenhaus bzw. dem Eingriffsdatum bei ambulanten Fällen.

Um die komplexen Anforderungen an die sektorenübergreifende QS-Dokumentation sowie die zugehörigen Datenflüsse zu erfüllen, besteht die Spezifikation aus verschiedenen Komponenten, die je nach Anwender spezifisch zusammengestellt werden. Als Komponenten werden dabei Access-Datenbanken, technische Dokumentationen, Ausfüllhinweise und anderes bezeichnet. Jeder Anwender bekommt damit das für ihn Relevante in einem eigenen Spezifikationspaket als Download zur Verfügung gestellt. Jedes dieser Pakete kann auf diese Weise auch unabhängig von den anderen aktualisiert werden.

Damit gibt es ein Spezifikationspaket für

- den Regelbetrieb
- ggf. Sonderexporte
- ggf. Probetriebe
- ggf. Testbetriebe

Sowohl die Spezifikationspakete als auch die einzelnen Komponenten werden nach einem einheitlichen Schema benannt, das bereits im Namen übersichtlich die relevanten Informationen wie Betriebsart, Exportformat und Versionierung enthält. Dieses Schema wird im nächsten Abschnitt detailliert erläutert. Durch die Versionierung sowohl auf der Ebene der Pakete als auch auf der Ebene der Komponenten ist gewährleistet, dass der aktuelle Stand leicht ersichtlich ist. Zudem wird die Kommunikation über die anzuwendenden Bestandteile der Spezifikation erleichtert.

Jedem Paket liegt eine Auflistung der einzelnen Komponenten und ggf. eine Übersicht über die Änderungen zur vorhergehenden Version bei.

### 1.1.1 Benennungsschema für Spezifikationspakete

Die Benennung der Spezifikationspakete setzt sich wie folgt zusammen:

<Erfassungsjahr>\_<Richtlinie>\_<Name>\_[<DAS>]\_<Betriebsart>\_<Exportformat>\_V<Versionsnummer>

Das Erfassungsjahr gilt für alle Spezifikationspakete und -komponenten, die Daten dieses Erfassungsjahres betreffen, egal in welchem Jahr das jeweilige Paket veröffentlicht wurde.

Da die Spezifikation der einrichtungsbezogenen QS-Dokumentation im Erfassungsjahr 2017 auf das Verfahren 2 (QS WI) der Richtlinie zur einrichtungs- und sektorenübergreifenden Qualitätssicherung beschränkt ist, steht im Platzhalter <Richtlinie> immer die Bezeichnung Qesue.

Bei der Angabe <Name> kann der die Spezifikation kennzeichnende Name angegeben werden. Namen können beispielsweise wie folgt definiert werden:

- FDOK: fallbezogene QS-Dokumentation
- EDOK: einrichtungsbezogene QS-Dokumentation
- SozDat: Nutzung von Sozialdaten bei den Krankenkassen
- DBSD: Datenformat zum Bericht über den Strukturierten Dialog

Bei der optionalen Angabe [ <DAS> ] kann beispielsweise zwischen folgenden Kürzeln unterschieden werden:

- LKG: LQS/LKG
- KV: DAS-KV
- KK: DAS-KK

Bei der Betriebsart kann zwischen folgenden Kürzeln unterschieden werden:

- RB: Regelbetrieb
- SE: Sonderexport
- PB: Probetrieb
- TB: Testbetrieb

V<Versionsnummer>: Die Versionierung erfolgt in ganzen Zahlen, die zweistellig angegeben sind (unter 10 mit einer vorlaufenden 0, z.B. V01).

---

### Beispiele:

Die einrichtungsbezogene QS-Dokumentation kann beispielsweise in folgendem Spezifikationspaket veröffentlicht werden:

2017\_Qesue\_EDOK\_RB\_XML\_V01

Die einrichtungsbezogene QS-Dokumentation (Dokumentation von Einrichtungen, die ambulante und stationäre Fälle am Krankenhaus behandeln) kann beispielsweise in folgendem Spezifikationspaket veröffentlicht werden:

2017\_Qesue\_EDOK\_LKG\_RB\_XML\_V01

Die einrichtungsbezogene QS-Dokumentation (Dokumentation von Einrichtungen, die durch die KV abgerechnete Fälle behandeln) kann beispielsweise in folgendem Spezifikationspaket veröffentlicht werden:

2017\_Qesue\_EDOK\_KV\_RB\_XML\_V01

---

### 1.1.2 Benennungsschema für Spezifikationskomponenten

Die Benennung der Spezifikationskomponenten lehnt sich an das bei den Spezifikationspaketen verwendete Prinzip an:

[<Erfassungsjahr>\_]<Art der Komponente>\_[<Exportformat>\_] [V<Versionsnummer>].<Dateierweiterung>

„Art der Komponente“ bezieht sich auf die jeweilige Funktion und wird durch ein Kürzel angegeben.

Die in eckige Klammern gesetzten Benennungsbestandteile sind optional und werden nur dann gesetzt, wenn dieses Merkmal für die Komponenten von Relevanz ist. Das Erfassungsjahr ist für Hilfsprogramme nicht relevant; bei diesen erfolgt die Versionierung mit einer Versionsnummer, die über die Jahre fortgeführt wird.

Die Bezeichnungen der einzelnen Komponenten werden in Abschnitt B Komponenten erläutert.

## 1.2 Zielsetzung und Zielgruppe

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation ist ein komplexes Regelwerk, das mithilfe verschiedener Komponenten verbindliche Grundlagen für alle Prozesse im Zusammenhang mit der Erfassung und Übermittlung von QS-Daten bei den unterschiedlichen Verfahrensteilnehmern (Leistungserbringer, Datenannahmestellen, Bundesauswertungsstelle) vorgibt und beschreibt. Dadurch werden die Bereitstellung valider und vergleichbarer Daten sowie ein unter datenschutzrechtlichen Gesichtspunkten sicherer Datenfluss gewährleistet.

## 1.3 Releaseplanung

Um Planungssicherheit zu gewährleisten und angemessen auf Fehler reagieren zu können, werden die Termine zur Veröffentlichung von Spezifikationspaketen (Versionen) eines Erfassungsjahres und zu Rückmeldefristen in der Technischen Dokumentation angekündigt. Die Releaseplanung der einrichtungsbezogenen QS-Dokumentation ist an jene der QS-Basispezifikation für Leistungserbringer angelehnt. Die im Folgenden genannten Meilensteine sind als Zielwerte zu betrachten und können im Regelbetrieb entsprechend der Vorgaben des G-BA, in Abstimmungen mit Verfahrensteilnehmern sowie Anforderungen aus der Umsetzung konkretisiert werden.

Frist	Meilenstein	Bereitstellung
nach Beschluss durch das Plenum des G-BA	2017 V01 Alpha (Alphaversion)	Veröffentlichung auf der IQTIG-Website ( <a href="http://www.iqtig.org">http://www.iqtig.org</a> )
Mai 2017	2017 V01 Beta (Betaversion)	Veröffentlichung auf der IQTIG- Website ( <a href="http://www.iqtig.org">http://www.iqtig.org</a> ) Frist für Rückmeldungen: 15.06.2017 (E-Mail an <a href="mailto:verfahrensupport@iqtig.org">verfahrensupport@iqtig.org</a> )
30.06.2017	2017 V01 (finale Version)	Veröffentlichung auf der IQTIG- Website ( <a href="http://www.iqtig.org">http://www.iqtig.org</a> ) Frist für Rückmeldungen: 15.09.2017 (E-Mail an <a href="mailto:verfahrensupport@iqtig.org">verfahrensupport@iqtig.org</a> )
30.09.2017	2017 V02 (Fehlerkorrekturen)	Veröffentlichung auf der IQTIG- Website ( <a href="http://www.iqtig.org">http://www.iqtig.org</a> )

Da die Dokumentation der einrichtungsbezogenen QS-Dokumentation erst nach Ablauf des Erfassungsjahres 2017 erfolgen kann, entsprechen die ICD-, OPS- und EBM-Kodes in der finalen

Spezifikationsversion bereits der für dieses Erfassungsjahr gültigen Version des DIMDI. Ein separates Update (DIMDI-Umstieg) entfällt daher.

Es erfolgt eine regelmäßige Abstimmung mit Verfahrensteilnehmern (z.B. Softwareherstellern, Datenannahmestellen) in Form von Informationstreffen, Workshops und Kommunikationsplattform. Zudem wurden neue Meilensteine in den Prozess der Systempflege integriert (z.B. die Bereitstellung von Änderungsempfehlungen und Alphaversionen) und Festlegungen getroffen, die die Qualität der Spezifikation erhöhen und die Richtlinienkonformität sicherstellen (z.B. werden wesentliche Änderungen nur im Rahmen finaler Versionen berücksichtigt).

### **Change- und Fehlermanagement**

Das IQTIG empfiehlt für die Optimierung der Zusammenarbeit mit den beteiligten Stellen die folgenden Aktivitäten:

- Meldung von festgestellten Fehlern (z.B. Spezifikations- und Softwarefehlern)
- Verbreitung von Änderungsvorschlägen
- Abstimmung von Terminen und Umsetzungen im Rahmen der Releaseplanung
- Erfahrungsaustausch, um eine möglichst einheitliche Vorgehensweise zu ermöglichen
- Abstimmung der Spezifikationsänderungen

Vorschläge, Fehlermeldungen und Diskussionspunkte können per E-Mail an den Verfahrenssupport oder über die Kommunikationsplattform mitgeteilt werden. Sollten Sie keine Zugangsdaten zur Kommunikationsplattform haben, obwohl Sie eine beteiligte Institution (z.B. Softwareanbieter, Datenannahmestelle) sind, lassen Sie sich bitte bei uns registrieren.

### **Ihr Ansprechpartner:**

Institut für Qualitätssicherung und Transparenz im Gesundheitswesen  
Katharina-Heinroth-Ufer 1  
10787 Berlin

Telefon: (+49) 30 58 58 26 340

Fax: (+49) 30 58 58 26 341

[verfahrenssupport@iqtig.org](mailto:verfahrenssupport@iqtig.org)

[www.iqtig.org](http://www.iqtig.org)

# A Prozesse

Im Folgenden werden die einzelnen Prozesse und Unterprozesse der Spezifikation beschrieben.

# 1 QS-Dokumentation

In diesem Kapitel werden die Prozessschritte sowie die in jedem Prozessschritt benötigten Werkzeuge der QS-Dokumentation in Bezug auf die Auslösung, Erfassung, Verarbeitung und Datenübermittlung der QS-Dokumentation beschrieben.

## 1.1 Auslösung

Für jeden Fall im Krankenhaus oder im vertragsärztlichen Bereich wird geprüft, welche Datensätze (Module) der externen Qualitätssicherung dokumentationspflichtig sind. Entspricht mindestens ein Fall (Tracer-Operation) den definierten Auslösekriterien, wird die einrichtungsbezogene QS-Dokumentation ausgelöst. Mit der vorliegenden Spezifikation können Systeme entwickelt werden, die eine zeit- und prozessnahe Auslösung von QS-Dokumentationen ermöglichen.

Der Algorithmus (Abschnitt B 1.4) trifft seine Entscheidung auf der Grundlage der medizinischen Routinedokumentationen bzw. Abrechnungsdaten (Kodierung der Diagnosen [Haupt- und Nebendiagnosen, Primär- und Sekundär Diagnosen], Prozeduren, Gebührenordnungspositionen gemäß EBM-Katalog).

**KH**

### Einrichtungen, die ambulante und stationäre Fälle am Krankenhaus behandeln

Die medizinischen Routinedokumentationen sind in jedem Krankenhausinformationssystem (KIS) verfügbar, da sie auch für die Umsetzung der Datenübermittlungsvereinbarung gemäß § 301 Abs. 3 SGB V (kurz: DÜV-301) nötig sind. Die Informationen werden in einzelnen Fällen auch über spezialisierte Systeme, die auf die individuellen Anforderungen im jeweiligen Bereich zugeschnitten sind, zur Auslösung herangezogen.

Der QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V ist in Abschnitt B 1.3.1 beschrieben.

**VÄ**

### Einrichtungen, die ambulante vertragsärztliche oder stationäre belegärztliche Leistungen erbringen

Für diese Einrichtungen (z.B. Arztpraxen) ist die Dokumentationspflicht über die bei den Kassenärztlichen Vereinigungen vorliegenden Abrechnungsdaten zu bestimmen. Im QS-Filter-Eingangsdatensatz ist definiert, welche Daten verwendet werden. Diese basieren auf der KVDT-Datensatzbeschreibung (ungeprüfter Eingangsdatensatz) der Kassenärztlichen Bundesvereinigung für den Einsatz von IT-Systemen in der Arztpraxis zum Zwecke der Abrechnung gemäß § 295 Abs. 4 SGB V.

Der QS-Filter-Eingangsdatensatz nach § 295 SGB V ist in Abschnitt B 1.3.1 beschrieben.

LE

## 1.2 Erfassung

Nach Auslösung eines Moduls durch den QS-Filter folgt der Prozess der Erfassung. Neben QS-Daten (Abschnitt 1.2.3) sind auch einrichtungsidentifizierende Daten wie das Institutionskennzeichen für das Krankenhaus oder die Betriebsstättennummer des niedergelassenen Leistungserbringers zu dokumentieren (Abschnitt 1.2.2). Die Erfassung ist abgeschlossen, wenn alle zu dokumentierenden Datenfelder unter Berücksichtigung von Abhängigkeiten und Plausibilitätsprüfungen vollständig erfasst wurden (Abschnitt 1.2.4). Die Regelungen zur Gestaltung von Eingabemasken sind Abschnitt 1.2.1 zu entnehmen.

### 1.2.1 Gestaltung von Eingabemasken

Die Benutzeroberfläche einer Erfassungssoftware (Graphical User Interface = GUI) oder einer webbasierten Lösung soll ergonomisch und anwenderfreundlich gestaltet sein. Gestaltung und Layout der Eingabemaske sind Aufgabe der Softwarehersteller bzw. der Webportal-Entwickler. Neben Anforderungen der Kunden werden üblicherweise firmeninterne Standards bzw. Vorgaben des Betriebssystems (z.B. Windows) für das „look and feel“ berücksichtigt.

Diese Spezifikation definiert als Minimalstandard die für den Anwender sichtbaren Inhalte der Dokumentationsformulare. Als Referenz für die sichtbaren Inhalte dienen die Dokumentationsbögen, die als Bestandteil der Spezifikation durch das IQTIG veröffentlicht werden. Die Dokumentationsbögen werden als PDF-Dokumente bereitgestellt, die aus der Spezifikationsdatenbank automatisch generiert worden sind.



#### Achtung

#### Verbindliche Grundlage zur Implementierung

Bei den Dokumentationsbögen handelt es sich um Formulare zur Ansicht (Muster), die nicht zur Dokumentation zu verwenden sind. Die Papierform ist hier nur als eine Abbildung eines denkbaren Eingabeformulars oder einer denkbaren Eingabemaske zu verstehen.

Verbindlich sind nur die Inhalte der Datenbank zur QS-Dokumentation.

Tabelle 1 gibt einen Überblick darüber, welche Informationen der Spezifikationsdatenbank (identifiziert durch Tabelle und Attribut) bei der Erstellung der Dokumentationsbögen berücksichtigt werden und somit auch in den Erfassungssystemen sichtbar sein müssen.

Tabelle 1: Informationen aus der Datenbank, welche im GUI verwendet werden

Tabelle	Attribut	Bemerkung	sichtbar für Anwender
Modul	name	Kürzel des Datensatzes (z.B. 18 / 1), erscheint üblicherweise im Titel des Formulars	ja

<b>Tabelle</b>	<b>Attribut</b>	<b>Bemerkung</b>	<b>sichtbar für Anwender</b>
Modul	bezeichnung	Bezeichnung des Datensatzes (z.B. Mammachirurgie), erscheint üblicherweise im Titel des Formulars	ja
Bogen	bezeichnung	Bezeichnung des Teildatensatzes	ja
Bogen-Feld	zeileAufBogen	Nummer des Datenfeldes, dient bei umfangreicheren Bögen zur besseren Orientierung	ja
Bogen-Feld	bezeichnung	Bezeichnung des Datenfelds	ja
Bogen-Feld	ergaenzendeBezeichnung	Ergänzende Bezeichnung zum Datenfeld, kann z.B. durch Wahl der Schrift von der Bezeichnung abgesetzt werden	ja
Feld	laenge	Definiert die Länge des Eingabefelds. Für die Gestaltung des Eingabefelds sind weitere Informationen aus der Datenbank wichtig (z.B. Feld.nachKommaLaenge).	ja
Feld	einheit	Einheiten (wie z.B. ml) müssen angezeigt werden.	ja
BasisTyp	format	Formatanweisungen (wie z.B. TT.MM.JJJJ) sollen –sofern nicht durch übergeeignete Eingabefelder unterstützt– für den Anwender angezeigt werden.	ja
SchlüsselWert	code	Bei Schlüsselfeldern sollen die Codes möglichst in Auswahllisten angezeigt werden. Bei einigen Realisierungsvarianten (z.B. Checkbox) kann auf die Anzeige der Codes verzichtet werden.	(ja)
SchlüsselWert	bezeichnung	Bei Schlüsselfeldern müssen die Textdefinitionen der Codes (z.B. in einer Auswahlliste) angezeigt werden. Für die Sortierung sind die Attribute <code>sortierNrVerwendet</code> und <code>zahl der Tabelle Schlüssel</code> relevant (Abschnitt B 2.2.3).	ja

Tabelle	Attribut	Bemerkung	sichtbar für Anwender
Ab-schnitt	bezeichnung	Die Überschriften sind wichtig für die Strukturierung und das Verständnis des Datensatzes und müssen deshalb in der QS-Dokumentationssoftware angezeigt werden.	ja

Werden Datenfelder (z.B. OPS) eines QS-Datensatzes aus Fremdsystemen über Schnittstellen importiert, so sollen die übernommenen Daten auch in der Erfassungssoftware angezeigt werden. Es ist für den Anwender wichtig, die vollständigen QS-Daten im Kontext eines „QS-Formulars“ zu sehen und auch auf Richtigkeit und Vollständigkeit zu prüfen.

### Allgemeine Grundsätze für die Gestaltung der Eingabemaske

Die Grundsätze für die Plausibilitätsprüfungen wirken sich insbesondere auf die Gestaltung der Benutzeroberflächen in Erfassungssoftware oder Weblösung aus. Durch die funktionale Gestaltung sollte ein Kompromiss zwischen Dateneingabekomfort einerseits und Zwang zur aktiven Eingabe korrekter Daten andererseits gefunden werden. Im Folgenden werden die Regeln für die Gestaltung von Benutzeroberflächen aufgeführt:<sup>3</sup>

- Keine Suggestion von Feldinhalten durch Vorbelegung (Defaults): Oberstes Prinzip bei der Gestaltung der Benutzeroberflächen ist, dass dem Anwender des Programms keine Angaben suggeriert werden. Insbesondere darf keine Vorbelegung mit Standardwerten erfolgen, die den „Nicht-Problemfall“ dokumentieren.



#### Hinweis

Eine Vorbelegung mit Standardwerten ist nicht zulässig. Die Übernahme von im KIS/AIS vorhandenen Angaben in die QS-Dokumentation ist hingegen zulässig. Beispielsweise kann die Angabe des Datenfeldes IKNRKH aus dem KIS übernommen werden, wenn diese Angabe vorliegt. Die Frage, ob während des stationären Aufenthaltes Komplikationen aufgetreten sind, darf hingegen nicht mit einem Standardwert „nein“ vorbelegt werden, wenn hierzu keine Angaben im KIS/AIS enthalten sind.

- Verwendung der vorgeschriebenen Fehler- und Warnmeldungen bei feldübergreifenden Regeln: Die Fehler- und Warnmeldungen sind so formuliert, dass sie möglichst nicht suggerieren, auf welche Weise widersprechende Angaben korrigiert werden sollen. Insofern müssen sie wörtlich übernommen werden.

<sup>3</sup> Externe Systeme, die Daten an ein Erfassungsprogramm übergeben, sollten diese Grundsätze sinngemäß anwenden.



### **Achtung** **Verbindliche Grundlage zur Implementierung**

Plausibilitätsregeln sind nicht in allen Fällen den Dokumentationsbögen zu entnehmen. Verbindlich sind nur die Inhalte der Datenbank zur QS-Dokumentation.

---

- Keine zusätzlichen Ober-/Untergrenzen für Maße, Zeitdauern und Anzahlen: Außer den durch die Datenfeldbeschreibung und die Plausibilitätsregeln vorgegebenen Wertebereichen darf in Erfassungsprogrammen keine Einengung möglicher Merkmalsausprägungen in Wertefeldern erfolgen.
  - Zwang zur aktiven Entscheidung zwischen „ja“ und „nein“: An entsprechenden Stellen in den Dokumentationsbögen, bei denen die Auswahl „0“ (nein) und „1“ (ja) (vgl. z.B. Schlüssel JN) zu treffen ist, darf keine Voreinstellung des Wertes im Eingabefeld erfolgen. Es besteht somit der Zwang zur Eingabe eines Wertes. Nur an Stellen, an denen im Erfassungsformular lediglich „1“ als Option angegeben wird, soll die Nicht-Eingabe eines Wertes als Verneinung interpretiert werden. Hintergrund dieser Differenzierung ist, dass einerseits in qualitätskritischen Bereichen eine Unterscheidung zwischen „keine Angabe“ und „nein“ erfolgen muss, es andererseits der Benutzerakzeptanz abträglich ist, wenn diese Systematik auch an allen anderen Stellen durchgängig verfolgt wird.
  - Kein automatisches Verändern von Feldinhalten in Abhängigkeit von anderen Feldinhalten.
- 

#### **Beispiel<sup>4</sup>:**

Wenn die Transfusion von Blut zunächst bejaht und das Feld „Fremdblut“ angekreuzt worden ist, soll das Entfernen des Kreuzes des übergeordneten Feldes „Bluttransfusion“ nicht automatisch zum Entfernen des Kreuzes bei „Fremdblut“ führen. Vielmehr soll eine Fehlermeldung erfolgen und der Anwender gezwungen sein, zunächst das Kreuz bei „Fremdblut“ zu entfernen, bevor „Bluttransfusion“ verneint werden kann.

---

#### **Umrechnung von Einheiten bei numerischen Feldern**

In Einzelfällen ist es aus Anwendersicht hilfreich, wenn die -Eingabemaske die Eingabe von Messwerten in Einheiten ermöglicht, die von den spezifizierten Einheiten abweichen. Beispielsweise soll im Modul NWIES laut Spezifikation das Datenfeld DESINFEKTIONOP in Litern mit einer Nachkommastelle dokumentiert werden. Die Funktionalität einer automatischen Umrechnung sollte möglichst in die Erfassungssoftware oder die Weblösung integriert werden, um den Dokumentationsaufwand zu verringern.

Exportiert werden die Datenfelder in der gemäß Spezifikation gültigen Einheit.

#### **Gestaltung von Eingabemasken mit Layout-Feldgruppen**

Sogenannte Layout-Feldgruppen sollen in der Erfassungsmaske separat kenntlich gemacht werden. Hierbei handelt es sich um Datenfelder, die zu einer logischen Gruppe zusammengefasst werden können. Feldgruppen können Filterfelder oder abhängige Felder beinhalten. Abhängige

---

<sup>4</sup> Die im Beispiel genannten Felder sind NICHT Bestandteil der Spezifikationsdatenbank für die QS-Dokumentation. Sie sind frei erfunden und dienen lediglich der Erläuterung.

Felder von Layout-Feldgruppen werden auf den generierten Dokumentationsbögen eingegraut dargestellt. Hierbei handelt es sich um Feldgruppen, bei denen das Attribut `grauWennNegativ` in der Datenbanktabelle `FeldGruppe` auf `TRUE` gesetzt ist. Diese sollen für die Gestaltung von Eingabemasken verwendet werden.

Die Eigenschaften von Layout-Feldgruppen sind in Abschnitt B 2.3.7 erläutert.

### **Empfehlung zur Umsetzung von Layout-Feldgruppen**

Ist das Attribut `grauWennNegativ` gesetzt, so darf die Benutzereingabe für die abhängigen Felder durch die Erfassungssoftware deaktiviert werden, falls die negative Filterbedingung zutrifft. Bei der Umsetzung muss Folgendes sichergestellt werden:

- Nach jeder Änderung der Inhalte der Filterfelder im Erfassungsformular muss das Programm die Filterbedingung der Feldgruppe evaluieren und ggf. eine Aktualisierung der Oberfläche durchführen.
- Die Benutzereingabe für die abhängigen Felder darf nur dann deaktiviert werden, wenn keines dieser Felder ausgefüllt ist. Ansonsten ist der Anwender auf eine Plausibilitätsverletzung hinzuweisen.
- Wenn nach einer Benutzereingabe die positive Filterbedingung zutrifft, so sind ggf. vorher deaktivierte Eingabefelder wieder zu aktivieren.
- Deaktivierte Felder dürfen nicht ausgeblendet werden.

## **1.2.2 Einrichtungsidefizierende Daten**

In Hinblick auf eine einrichtungsbezogene Auswertung und Berichterstattung sind einrichtungsidentifizierende Daten in der QS-Dokumentation zu erfassen.

**KH**

### **Einrichtungsidefizierende Daten des Krankenhauses**

Wichtigstes Merkmal zur Identifikation der Einrichtung ist das Institutionskennzeichen.

Im sektorenübergreifenden Verfahren QS WI werden belegärztliche Eingriffe nicht dem Krankenhaus, sondern dem Belegarzt als Vertragsarzt zugeordnet.

Belegärztliche Fälle werden in diesem Verfahren über eine durch die Kassenärztlichen Vereinigungen zur Verfügung gestellte, z.B. webbasierte Lösung erfasst, die nicht das Institutionskennzeichen erfordert, sondern das einrichtungsidentifizierende Datum der Arztpraxis des Belegarztes.

**VÄ**

### **Einrichtungsidefizierende Daten der Arztpraxis bzw. des niedergelassenen Leistungserbringers**

Die wesentliche Angabe zur Identifikation eines ambulanten Leistungserbringers ist die von der Kassenärztlichen Vereinigung (KV) vergebene Betriebsstättennummer (BSNR; Feld `BSNRAMBULANT`). Sie identifiziert die Arztpraxis als abrechnende Einheit. Dabei umfasst der Begriff Arztpraxis auch Medizinische Versorgungszentren (MVZ), Institute, Notfallambulanzen sowie Ermächtigungen von am Krankenhaus beschäftigten Ärzten. Der Ort der Leistungserbringung ist dabei nachrangig: Die Leistung kann auch im Rahmen einer KV-Ermächtigung von einem Krankenhausarzt oder als belegärztlicher Eingriff durch den niedergelassenen Leistungserbringer am

Krankenhaus durchgeführt worden sein. Im sektorenübergreifenden Verfahren QS WI sind belegärztliche Leistungen mit der Betriebsstättennummer der ambulanten Einrichtung des Belegarztes zu kennzeichnen.

### 1.2.3 QS-Daten

Es sind alle erforderlichen Daten zu dokumentieren. Hierbei kann eine automatische Übertragung der QS-Daten aus dem KIS/AIS möglich sein.

### 1.2.4 Plausibilitätsprüfungen

Fehlende und widersprüchliche Angaben in den Datensätzen sollen durch umfangreiche Plausibilitätsprüfungen verhindert werden. In der QS-Dokumentation muss die vollständige Plausibilitätsprüfung für jeden Datensatz spätestens bei Dokumentationsabschluss erfolgen. Teile der Plausibilitätsprüfungen sollen bereits während der Erfassung erfolgen. Dadurch wird sichergestellt, dass ein aufwendiges Korrekturverfahren weitgehend entfällt.

Es gelten folgende Grundsätze für die Plausibilitätsprüfung:

- Alle Felder müssen ausgefüllt sein, wenn andere logische Sachverhalte dem nicht entgegenstehen.
- Jedes Feld, das auszufüllen ist, muss einen sinnvollen Feldinhalt haben.
- Es wird jede harte Plausibilitätsprüfung vorgenommen, die definiert ist.
- Harte Plausibilitätsprüfungen werden nur vorgenommen, wenn Sachverhalte zwingend miteinander gekoppelt sind.
- Es werden keine Sachverhalte suggeriert (keine Default-Werte, keine Vorbelegungen, keine Profile. Fehlermeldungen werden vorgegeben).
- Keine Angabe (bzw. kein Feldinhalt) wird ergänzt oder gelöscht.

#### Arten der Plausibilitätsprüfungen

Es wird zwischen folgenden Arten von Plausibilitätsprüfungen unterschieden:

- **harte Prüfungen:** Bei einer harten Regelverletzung ist ein Dokumentationsabschluss der QS-Dokumentation eines Vorgangs unzulässig. Die in der technischen Dokumentation und der Datenbank definierten Plausibilitätsprüfungen sind hart, außer wenn sie explizit als weich gekennzeichnet sind.
- **weiche Prüfungen:** Die weichen Plausibilitätsprüfungen der Spezifikation sind bis spätestens zum Dokumentationsabschluss durchzuführen. Bei einer Regelverletzung erhält der Benutzer einen Warnhinweis, anhand dessen er entscheidet, ob eine Änderung von Feldinhalten notwendig ist.

#### Einzelregeln

Einzelregeln sind in der Datenbank zur QS-Dokumentation in Regelsyntax in der Tabelle `Regeln` hinterlegt. Außerdem gibt es Regeln, die nur in Form von Feldeigenschaften – nicht aber in Regelsyntax – in der Datenbank hinterlegt sind. Die standardisierten Fehlertexte dieser Prüfungen sind Abschnitt B 2.3.8 zu entnehmen.

Es müssen sowohl die harten als auch die weichen feldbezogenen Prüfungen ausgeführt werden. Die Evaluation soll direkt bei der Dateneingabe geschehen. Fehleingaben sollen dem Benutzer direkt mitgeteilt werden. Einige Prüfungen erübrigen sich durch adäquate Gestaltung von Eingabemasken, z.B. durch Bereitstellung von Auswahlmenüs für Schlüsselkodes.

**Feldgruppenregeln**

Datenfelder (Bogenfelder) eines Moduls können zu einer Feldgruppe zusammengefasst werden, um logische Abhängigkeiten von Bogenfeldern abzubilden. Das bedeutet in der Praxis, dass der Anwender daran gehindert wird, Felder mit Werten auszufüllen, die der Logik der Feldgruppe widersprechen.

Die explizite Definition von Feldgruppen strukturiert sowohl die Bogenfelder als auch die Plausibilitätsregeln, indem diese die Bogenfelder eines Moduls zu einer logisch zusammenhängenden Gruppe von Feldern zusammenfassen. Die Feldgruppen ergeben sich dabei indirekt aus der Definition von Plausibilitätsregeln.

Die Abbildung von Feldgruppenregeln in der Datenbank für QS-Dokumentation ist in Abschnitt B 2.3.7 erläutert.

**1.3 Export der Daten**

Die dokumentierten QS-Daten müssen bei sektorenübergreifenden Verfahren gem. Qesü-RL in Abhängigkeit vom Abrechnungskontext der Leistungserbringung verschiedenen Datenflüssen zugeordnet werden.

Abbildung 1 stellt das serielle Datenflussmodell gem. Qesü-RL dar. Hierbei sind je nach Leistungserbringer bzw. Abrechnungskontext unterschiedliche Datenannahmestellen vorgesehen:

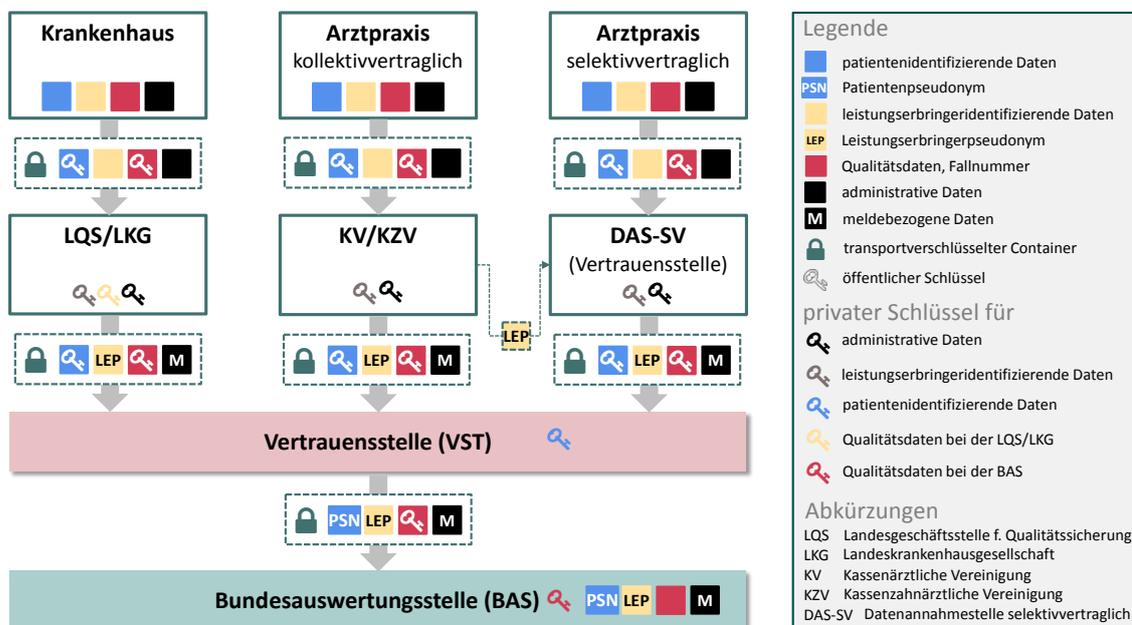


Abbildung 1: Serielles Datenflussmodell für die sektorenübergreifende Qualitätssicherung

Die Zuordnung der einzelnen Fälle zu den Leistungserbringern gemäß Datenfluss nach Qesü-RL richtet sich primär nach dem Abrechnungskontext und nicht nach dem Ort der Leistungserbringung.



#### **Hinweis zum Datenfluss**

Die einrichtungsbezogenen QS-Daten nach § 3 Absatz 2 der Qesü-RL werden direkt von der Datenannahmestelle an die Bundesauswertungsstelle weitergeleitet, da keine Pseudonymisierung durch die Vertrauensstelle erfolgt.

---

Die Exportdateien werden beim Leistungserbringer erstellt und dann an die zuständige Datenannahmestelle (DAS) versendet. Datenannahmestellen nehmen die Datenlieferungen im Rahmen der Qesü-RL entgegen, führen die Leistungserbringerpseudonymisierung durch und leiten anschließend die Daten weiter an die Bundesauswertungsstelle. Der Export und die Übermittlung sämtlicher Daten finden im XML-Format statt.

#### **1.3.1 Erzeugen der Exportdatei**

Die zu exportierenden Daten der einrichtungsbezogenen QS-Dokumentation werden vom Dokumentationssystem in Exportdateien geschrieben und die entsprechenden Vorgänge (identifiziert durch Vorgangsnummern<sup>5</sup>) im absendenden Dokumentationssystem als „exportiert“ markiert.

#### **Inhalte der Exportdatei**

Aufgrund der einheitlichen Leistungserbringerpseudonymisierung dürfen nur Datensätze der Module gemeinsam exportiert werden, die einem gemeinsamen Pseudonymisierungsverfahren unterliegen.

---



#### **Hinweis zum Datenfluss**

Die Datensätze der fall- und einrichtungsbezogenen QS-Dokumentation können aufgrund der unterschiedlichen Datenflüsse nicht gemeinsam exportiert werden, obwohl sie demselben Pseudonymisierungsverfahren unterliegen.

Die einrichtungsbezogenen QS-Daten nach § 3 Absatz 2 der Qesü-RL werden direkt von der Datenannahmestelle an die Bundesauswertungsstelle weitergeleitet, da keine Pseudonymisierung durch die Vertrauensstelle erfolgt. Der Datenfluss für die fallbezogenen QS-Daten (NWIF) erfolgt hingegen unter Einbeziehung einer Datenannahmestelle und einer Vertrauensstelle an die Bundesauswertungsstelle.

Dateien, in denen diese Trennung nicht eingehalten wird, werden von der Datenannahmestelle zurückgewiesen.

---

Folgende Module des Verfahrens QS WI unterliegen dem gemeinsamen Pseudonymisierungsverfahren NWI:

---

<sup>5</sup> Datensatznummern.

Tabelle 2: Pseudonymisierungsverfahren des Verfahrens QS WI

Richtlinie	Bezeichnung	Erhebungsinstrument	Verfahrenskennung	Exportmodul	VST	DAS
Qesü	Vermeidung nosokomialer Infektionen: Postoperative Wundinfektionen (einrichtungsbezogen ambulant) (NWIEA)	Einrichtungsbezogene QS-Dokumentation	NWI	NWIEA_KV NWIEA_LKG		X
	Vermeidung nosokomialer Infektionen: Postoperative Wundinfektionen (einrichtungsbezogen stationär) (NWIES)		NWI	NWIES_KV NWIES_LKG		X
	Vermeidung nosokomialer Infektionen: Postoperative Wundinfektionen (fallbezogen) (NWIF)	Fallbezogene QS-Dokumentation	NWI	NWIF	X	X
	Nosokomiale Wundinfektionen - Postoperative Wundinfektionen (NWIWI)	Erhebung von Sozialdaten bei den Kassen	NWI	NWIWI	X	X
	Nosokomiale Wundinfektionen – Tracer (NWITR)		NWI	NWITR		X

### Export von Teildatensätzen

Beim Export einer QS-Dokumentation durch ein Dokumentationssystem werden die Inhalte der für den betreffenden Vorgang angelegten Teildatensätze hierarchisch in den XML-Code des passenden Basisbogens geschrieben und können nur gemeinsam mit dem Inhalt des Basisbogens exportiert werden. Die Struktur der Einbettung ist durch den Datentyp des Exportmoduls im Schema definiert.

Die Reihenfolge der Unterbögen (Teildatensätze) ist in der Spezifikationsdatenbank festgelegt. In der Tabelle Bogen ist in der Spalte *SortierNr* die Reihenfolge der Unterbögen bei der Erfassung und beim Export aufgeführt.

## Anonymisierung

Die Vorschriften zur Anonymisierung von Bogenfeldern sind in den Tabellen `ErsatzFeld` und `ErsatzFuerFeld` in der Spezifikationsdatenbank für QS-Dokumentation zu finden (Abschnitt B 2.4.3). Da im Rahmen der einrichtungsbezogenen QS-Dokumentation keine personenbezogenen Daten erhoben werden, ist eine Anonymisierung durch die QS-Dokumentationssoftware derzeit nicht notwendig.

## Aufbau der Exportdatei

Die innere Struktur der Exportdatei ergibt sich unter anderem aus der Datenfeldbeschreibung der einzelnen Module und den oben beschriebenen Anonymisierungsmaßnahmen zur Erfüllung datenschutzrechtlicher Vorgaben. Aufbauend auf dieser Beschreibung wird ein XML-Schema abgeleitet. Die Struktur der Exportdatei wird durch entsprechende XML-Schemata festgelegt (Kapitel B 3).

Das XML-Schema beschreibt und definiert die Struktur des XML-Dokuments (Exportdatei) sowie den Inhaltstyp (Datentypen der einzelnen Bögen und Felder).

Die Exportdateien sind wie folgt aufgebaut:

- Header-Bereich enthält Metadaten (administrative Daten)
- Body-Bereich enthält die einrichtungsidentifizierenden (1.2.2) und die QS-Daten (1.2.3)



### Hinweis

Es ist zu beachten, dass die Zeichenkodierung deklariert wird und diese der UTF-8-Kodierung entsprechen muss.

---

## Sonderzeichen in XML

Folgende Zeichen, die bei der XML-Syntax besondere Bedeutung haben, müssen geschützt werden, falls Sie als Inhalte zu übermittelnder Informationen (Attributwerte) vorkommen sollen:

- `&` `&amp;`
- `<` `&lt;`
- `>` `&gt;`

Folgende Zeichen (Apostroph bzw. Anführungszeichen) müssen nur dann entsprechend geschützt werden, wenn sie selbst zur Übermittlung von Attributwerten verwendet werden.

- ``` `&apos;` (muss nur dann geschützt werden, wenn das Apostroph selbst zur Übermittlung von Attributwerten verwendet wird.)
- `"` `&quot;` (muss nur dann geschützt werden, wenn das Anführungszeichen selbst zur Übermittlung von Attributwerten verwendet wird.)

## Felder der Exportdatei

Einen Überblick über die zu exportierenden Felder eines Moduls liefert die Abfrage `ExportFelderFürEinModul`.<sup>6</sup>

---

<sup>6</sup> Metadaten des Datencontainers sind nicht Teil der Spezifikationsdatenbank und werden daher nicht durch die Abfrage ausgegeben.

### Export von Listenfeldern

Alle Elemente von Listenfeldern werden exportiert, ohne die Nummer des Listenfelds (im Namen des Exportfelds) an den Namen des Listenfelds anzuhängen (zu Listenfeldern siehe Abschnitt B 2.2.3).



#### Hinweis

In der vorliegenden Spezifikation zur einrichtungsbezogenen QS-Dokumentation sind derzeit keine Listenfelder vorgesehen.

---

### Zusatzfelder des Datenexports

Zusatzfelder und administrative Felder im Header, die nicht in der Datenfeldbeschreibung (Tabelle BogenFeld) eines Moduls enthalten sind, sind von der QS-Dokumentationssoftware zu füllen.<sup>7</sup>

Die Zusatzfelder sind in der Tabelle ZusatzFeld definiert. Das übertragene Speicherdatum DokAbschlDat (Datum des Dokumentationsabschlusses bzw. der Freigabe des Datensatzes für den Export) ist nicht Teil der Datenbank für Auswertungen und wird nur für organisatorische Zwecke verwendet.

Bei der Organisation im XML-Dokument ist weiter die Abfrage vExportZieleXml aus den administrativen Objekten zu berücksichtigen (Abschnitt B 2.6). Die Informationen in dieser Abfrage schließen aus, dass bestimmte Felder dem Element <qs\_data> zugeordnet werden, indem anhand eines XPath-Ausdrucks (xmlXPath) auf die richtige Stelle im XML-Dokument verwiesen wird.

### Organisation der Exportfelder im XML-Dokument

Die Exportfelder sind abhängig von der Modul- und Teildatensatzzugehörigkeit des Datensatzes im Element <qs\_data>, einem Unterelement des Elements <case> unterzubringen (Element qs\_data, Kapitel B 3).

### Stornierung

Um den Datensatz zu stornieren, muss <case>/<case\_admin><action> auf „delete“ gesetzt werden. Die Datenannahmestelle wird dadurch veranlasst, den betreffenden Datensatz einschließlich aller Vorversionen und Teildatensätze als „storniert“ zu kennzeichnen. Der Stornovorgang wird in der Datenbestätigung protokolliert.

Der zu stornierende Datensatz muss ebenfalls eine hochgezählte/fortgeschriebene Versionsnummer enthalten, um die Stornierung unabhängig von der Reihenfolge der Verarbeitung von Datensätzen sicherzustellen. Ein Storno mit einer bereits verwendeten Versionsnummer wird zurückgewiesen (Bestätigungsstatus ERROR, Fehlerart DOPPELT). Ein Stornoversuch eines noch nicht übermittelten Datensatzes wird ebenfalls zurückgewiesen.

---

<sup>7</sup> Hier gilt demnach nicht der Grundsatz, dass Felder nicht vorbelegt sein dürfen.

Zur Stornierung eines Datensatzes (Vorgang) genügt der Export der entsprechenden administrativen Daten `<case>/<case_admin>`. Die QS-Daten (`<qs_data>`) des zu stornierenden Datensatzes sind nicht erneut zu übermitteln.

### 1.3.2 Datenprüfung

Zusätzlich zu bereits im Rahmen der Erfassung durchzuführenden feldbezogenen und -übergreifenden Plausibilitätsprüfungen kann nach dem Datenexport die gesamte Struktur der XML-Datei durch aus der Spezifikationsdatenbank abgeleitete Schemata geprüft werden (Kapitel B 3). Welches Schema für einen Leistungserbringer vorgesehen ist, zeigt Tabelle 3

Tabelle 3: XML-Schemata zur Prüfung vor der Verschlüsselung

Richtlinie	Bereich	Schnittstelle	Schema
Qesü	Krankenhaus	interface_LE	interface_LE_lqs.xsd
	kollektivvertraglich	interface_LE	interface_LE_kv.xsd

Die einfachste Lösung für die Prüfung der Datenstruktur und der Plausibilität liegt in der Nutzung eines Datenprüfprogramms, das auf der Basis von XSLT die aus der Spezifikationsdatenbank QSDOK ausgeleiteten Plausibilitätsregeln in dem XML-Dokument prüft und Verstöße entsprechend im XML-Code dokumentiert. Ein solches Datenprüfprogramm setzt einen Standard für die Güte der Daten, der unbedingt einzuhalten ist. Die Datenprüfung muss an der Exportdatei vor der nachfolgend beschriebenen XML-Verschlüsselung vorgenommen werden.

Das Datenprüfprogramm ersetzt jedoch nicht die Verpflichtung der Softwareanbieter bzw. der Webportal-Entwickler<sup>8</sup>, schon bei der Eingabe der Daten eines Datensatzes, d.h. dokumentationsbegleitend, die Einhaltung der Plausibilitätsregeln sicherzustellen.

Details zur Verwendung eines Datenprüfprogramms können der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer bzw. Abschnitt B 4.2 entnommen werden.

### 1.3.3 XML-Verschlüsselung

Nach Fertigstellung des Dokuments und seiner Prüfung mit dem Datenprüfprogramm müssen einzelne Datenbereiche verschlüsselt werden. Die Verschlüsselung erfolgt asymmetrisch mit einem öffentlichen Schlüssel. Die öffentlichen Schlüssel der Datenannahmestellen (LQS/LKG) und der Bundesauswertungsstelle stehen den Anwendern auf der IQTIG-Website<sup>9</sup> zur Verfügung.

Diese Verschlüsselung baut auf dem W3C-XML-Encryption-Standard auf. Das IQTIG stellt ein Verschlüsselungsprogramm bereit, mit dem eine verfahrenskonforme XML-Verschlüsselung

<sup>8</sup> Die Kassenärztlichen Vereinigungen als DAS stellen den Vertragsärzten in ihrem Zuständigkeitsbereich Anwendungen zur Erfassung und Übertragung von Daten der einrichtungsbezogenen QS-Dokumentation, z.B. webbasiert, zur Verfügung (ggf. mit Unterstützung durch die Kassenärztliche Bundesvereinigung). Im Krankenhaussektor wird von Softwareherstellern eine QS-Software umgesetzt.

<sup>9</sup> <https://iqtig.org/datenerfassung/datenerfassung-und-uebermittlung/servicedateien/>

nach diesem Standard durchgeführt werden kann. Das Einbinden der Funktionen des Verschlüsselungsprogramms erfolgt u.a. über einen Befehlszeilenaufruf mit Parametern.

Tabelle 4 zeigt, in Abhängigkeit (erste Spalte) von der Stelle im Datenfluss (LE, DAS) und dem Weg des Datenflusses (über KV oder über LQS/LKG), welche Datenbereiche (Kopfzeile) mit welchem öffentlichen Schlüssel verschlüsselt werden müssen.

Tabelle 4: Asymmetrische Verschlüsselung der XML-Elemente

	/qs_data	/feedback_key	/care_provider
Verschlüsselung beim LE vor Versand an KV	mit öffentlichem Schlüssel der BAS	mit öffentlichem Schlüssel der BAS	keine Verschlüsselung
Verschlüsselung beim LE für Versand an LQS/LKG	mit öffentlichem Schlüssel der LQS/LKG	Element nicht vorhanden	keine Verschlüsselung
Verschlüsselung durch KV nach LE-Pseudonymisierung vor Weiterleitung	Element liegt bereits verschlüsselt vor	Element liegt bereits verschlüsselt vor	mit öffentlichem Schlüssel der BAS
Verschlüsselung durch LQS/LKG nach LE-Pseudonymisierung vor Weiterleitung	Mit öffentlichem Schlüssel der BAS	Element nicht vorhanden	Mit öffentlichem Schlüssel der BAS

### 1.3.4 Ausgangsvalidierung

Als letzte Maßnahme vor der Weiterleitung muss das Dokument gegen das Übertragungsschema auf Gültigkeit geprüft werden. Durch diese Prüfung wird sichergestellt, dass die richtigen Bereiche des XML-Codes verschlüsselt sind, und sie schließt aus, dass kritische Daten wesentlich unverschlüsselt den Leistungserbringer verlassen.

Das für einen bestimmten Leistungserbringer geltende Schema kann der gesonderten Beschreibung der Schema-Familie entnommen werden (Abschnitt 3.2 Schnittstellen).

Tabelle 5: XML-Schemata zur Prüfung nach der Verschlüsselung

Richtlinie	Bereich (Versand von)	Schnittstelle	Schema
Qesü	Krankenhaus (LE)	interface_LE_DAS	interface_LE_LQS
	Kollektivvertraglich (LE)	interface_LE_DAS	interface_LE_KV
	KV (DAS)	interface_DAS_BAS	interface_DAS_BAS
	LQS/LKG (DAS)	interface_DAS_BAS	



### Hinweis

Die für die Ausgangsvalidierung zu verwendenden Schemata sind dieselben, wie die bei der Eingangsvalidierung der DAS verwendeten Schemata. Bei nicht bestandener Schemaprüfung wird die Annahme des Dokuments verweigert.

### 1.3.5 Beispiele für Exportdateien

Beispieldateien unterschiedlicher Schnittstellen können der Spezifikationskomponente „XML\_Schema“ entnommen werden. Im Ordner XML-Beispiele sind für alle Schnittstellen entsprechende Ordner hinterlegt, deren Beispieldateien jeweils im Unterordner „export“ abgelegt sind. Die Benennung der Beispieldateien ist an die Benennung der zugehörigen Schemata angelehnt.

## 1.4 Datenübermittlung

Im folgenden Abschnitt werden die abschließend zu unternehmenden Arbeitsschritte Dateibenennung und Datenversand beschrieben.

Vor der Datenübermittlung muss die XML-Verschlüsselung erfolgreich durchgeführt worden sein. Ob sie erfolgreich durchgeführt wurde, kann mit den entsprechenden Schemata für die Ausgangskontrolle geprüft werden, vgl. Abschnitt 3.2 Schnittstellen.

Für die sektorenübergreifenden Verfahren werden die Datenlieferfristen in den Themenspezifischen Bestimmungen der **Qesü-RL** festgelegt. Sie werden vom IQTIG auf der Webseite [www.iqtig.org](http://www.iqtig.org) veröffentlicht.

Die Übermittlung von Daten muss grundsätzlich gesichert sein, unabhängig von der XML-Verschlüsselung gezielter, einzelner Inhalte. Diese gesicherte Übermittlung kann einerseits über gesicherte Übertragungswege (z. B. durch eine SSL-Verschlüsselung einer Webschnittstelle oder Übertragung mittels SFTP) gewährleistet werden sowie alternativ über eine zusätzliche Sicherung der zu übertragenden Informationen (Verschlüsselung der Pakete).

### 1.4.1 Dateibenennung

Die Exportdatei muss nach einem bestimmten Schema benannt werden.

---

#### Beispiel:

QS-Daten eines LE

47d16341-9e27-4e75-a27e-b791fbbd2dc8\_Q\_LE.xml

---

### 1.4.2 Datenversand via E-Mail

Für die Übermittlung der Daten über nicht gesicherte Übertragungswege (E-Mail) muss die Exportdatei in eine Transaktionsdatei (verpackte und symmetrisch verschlüsselte AES-Datei) umgewandelt werden.

Die Erstellung der Transaktionsdatei kann mit dem vom IQTIG bereitgestellten Verschlüsselungsprogramm (TPacker) durchgeführt werden. Das Einbinden der Funktionen des Verschlüsselungsprogramms erfolgt über einen Befehlszeilenaufwurf mit Parametern. Das Verschlüsselungsprogramm übernimmt auch die Dateibenennung der Transaktionsdatei mithilfe von übergebenen Parametern

**KH**

Die Voraussetzung für die Transportverschlüsselung zwischen LE und DAS ist die Registrierung der Dokumentationssoftware bei der zuständigen DAS. Bei der Registrierung erhält der Leistungserbringer einen Verschlüsselungskode, der bei der Erzeugung der Archivdatei verwendet werden muss.

**DAS**

Für die Übermittlung von DAS zu BAS ist ausschließlich das E-Mail-Verfahren vorgesehen. Die Übermittlung im Produktivbetrieb erfolgt an die E-Mail-Adresse [eb-daten@iqtig.org](mailto:eb-daten@iqtig.org). Eine Übermittlung im Testbetrieb wird an die E-Mail-Adresse [eb-daten-test@iqtig.org](mailto:eb-daten-test@iqtig.org) adressiert.

Die Transaktionsdatei wird nach einem vordefinierten Schema benannt (Abschnitt A 1.4 „Übermittlung der Daten im Datenfluss“):

---

**Beispiel:**

Transaktionsdatei

T-BU1234a-2016\_01\_19\_160945.zip.aes

T-BU1234a-2016\_01\_23\_114113\_045.zip.aes (millisekundengenau)

---

**1.4.3 Datenversand via gesicherter Schnittstellen – Arztpraxen/MVZ für kollektivvertraglich erbrachte Leistungen**

In der Regel wird derselbe Übertragungskanal wie für die Übertragung der Abrechnungsdaten, beispielweise KV-Connect, genutzt. KV-Connect wird als Anwendung im sicheren Netz der KVen (SNK) betrieben und ermöglicht eine sichere Ende-zu-Ende-Verschlüsselung sensibler und vertraulicher Patientendaten im Gesundheitswesen. Eine umfassende Sicherheitsanalyse<sup>10</sup> des TÜV Rheinland bestätigt, dass KV-Connect alle Kriterien des Anforderungskataloges zur Zertifizierung „Geprüfter Datenschutz“ erfüllt. Weiterhin wurden im Rahmen der Zertifizierung von KV-Connect als Anwendung im SNK die Anforderungen der Richtlinie an die KV-Applikationen<sup>11</sup> erfüllt. Diese Richtlinie orientiert sich an den Anforderungen des BSI Grundschutzkataloges.

**Hinweis**

Anders als bei der Datenübermittlung über nicht gesicherte Übertragungswege (E-Mail), muss die Exportdatei nicht mit dem Verschlüsselungsprogramm TPACKER transportverschlüsselt werden.

---

<sup>10</sup> <https://www.kv-telematik.de/partner-und-softwarehaeuser/weitere-infos/tuev-zertifizierung/>

<sup>11</sup> [http://www.kbv.de/media/sp/KBV\\_SNK\\_RLKV\\_KV\\_Apps\\_V3\\_0.pdf](http://www.kbv.de/media/sp/KBV_SNK_RLKV_KV_Apps_V3_0.pdf)

## 1.5 Rückprotokollierung

In diesem Kapitel wird die Rückprotokollierung in Bezug auf die Funktion, den Aufbau und die Erstellung beschrieben.

### 1.5.1 Funktion von Empfangsbestätigung und Datenflussprotokoll im Datenfluss

#### Empfangsbestätigung

Die Empfangsbestätigung wird nach Erhalt und abschließender erfolgreicher Eingangsverarbeitung und Weiterleitung eines Dokuments über den Eingangskanal<sup>12</sup> an den Absender übermittelt. Sie bestätigt dem Absender den Übergang der Verantwortung für das Dokument an den Aussteller.

Eine Empfangsbestätigung ist nur für den Absender bestimmt und wird nicht weitergeleitet. Bei Ausbleiben ist von einer fehlgeschlagenen Übermittlung auszugehen. Es gibt zurzeit keine verbindliche Vereinbarung, in welchem zeitlichen Rahmen eine Empfangsbestätigung erwartet werden kann. Angestrebt werden soll allerdings eine Echtzeitverarbeitung, sodass allein die Verarbeitungsdauer eines Dokuments die Verzögerung einer Empfangsbestätigung bedingt. Empfangsbestätigungen werden im Rahmen der einrichtungsbezogenen QS-Dokumentation lediglich von den DAS an die LE übermittelt. Empfangsbestätigungen von der BAS an die DAS sind aufgrund der Übermittlung eines Datenflussprotokolls (bzw. Miniprotokolls) nicht erforderlich.

---

#### Beispiel:

Wie die folgende Abbildung zeigt, enthält die Empfangsbestätigung lediglich eine Benachrichtigung, dass die vom Leistungserbringer übermittelte XML-Datei verarbeitbar war und an die nachfolgende Stelle weitergeleitet wurde:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" content_version="1.0" container_version="2.0"
xsi:noNamespaceSchemaLocation="../../../interface_LE_DAS/response_receipt.xsd"
xmlns:xenc="http://www.w3.org/2001/04/xmldsig#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <header>
    <document>
      <id V="{aaABa24e-bC3F-EeCe-Cb47-AddECBa96D1B}"/>
    </document>
    <provider email="datenannahmestelle@test-datenannahmestelle.de"
      name="Test-Datenannahmestelle" registration="bu00000"
      function="Datenannahmestelle" address="Test Adresse"/>
  </header>
  <body>
    <nachricht>
      Ihre Dateneinsendung konnte erfolgreich eingelesen und an die Bundesauswertungsstelle
      weitergeleitet werden. Ein umfassendes Datenflussprotokoll über die Prüfergebnisse
      der Datenannahmestelle und der Bundesauswertungsstelle erhalten Sie von uns in den nächsten 24 Stunden.
    </nachricht>
  </body>
</root>
```

Abbildung 2: Beispiel einer Empfangsbestätigung

---

<sup>12</sup> Es können Abweichungen auftreten. Nähere Informationen sind bei der zuständigen KV einzuholen.

Die Empfangsbestätigung soll ohne Verschlüsselung an den Sender geschickt werden. Für eine weitere Vereinfachung der Benachrichtigung können die Datenannahmestellen den Inhalt der spezifizierten Benachrichtigung z. B. in die E-Mail übernehmen (betreff/body). Die vollständige Benachrichtigung lautet:

*Ihre Dateneinsendung konnte erfolgreich eingelesen und an die Bundesauswertungsstelle weitergeleitet werden.*

*Ein umfassendes Datenflussprotokoll über die Prüfergebnisse der Datenannahmestelle und der Bundesauswertungsstelle erhalten Sie von uns innerhalb der nächsten 24 Stunden.*

Die Benennung der unverschlüsselten Datei leitet sich aus dem Benennungsschema ab (Abschnitt A 1.4.1).

### **Datenflussprotokoll**

Ein Datenflussprotokoll wird erstellt, wenn das Dokument keine weitere Verarbeitung mehr erlaubt. Das ist dann der Fall, wenn das Dokument durch einen der vorgesehenen Prüfungs- und Verarbeitungsschritte den Status `ERROR` erhält oder wenn das Dokument in der Bundesauswertungsstelle vollständig und erfolgreich verarbeitet wurde und den Status `WARNING` oder `OK` trägt.

Das Datenflussprotokoll dokumentiert alle an dem Dokument durchgeführten Prüfungen und deren Ergebnisse.

Ein Datenflussprotokoll wird in der Regel bis zum Leistungserbringer zurückübermittelt. Aus diesem Grund muss die Datenannahmestelle nach der Prüfung des erhaltenen Datenflussprotokolls den Leistungserbringer depseudonymisieren.

Die im Protokoll im Attribut „`originator`“ ggf. enthaltenen Hinweise auf den Verursacher eines Fehlers müssen von der Datenannahmestelle so ausgewertet werden, dass Verarbeitungsfehler, die weder auf Fehler des Leistungserbringers noch auf die Software/Weblösung des Leistungserbringers zurückzuführen sind, nicht an den Leistungserbringer weitergeleitet werden. Dieses ist daran erkennbar, dass im Attribut „`originator`“ ein von Leistungserbringer oder Softwareanbieter<sup>13</sup> abweichender Wert steht. In einem solchen Fall soll verhindert werden, dass das Datenflussprotokoll an den Leistungserbringer verschickt wird.

Statt der Weiterleitung sollte die fehlerhafte Verarbeitung des auslösenden Beteiligten (Datenannahmestelle, Bundesauswertungsstelle) korrigiert werden und ein korrekt verarbeitetes Dokument, dessen Stand in Abstimmung mit der Bundesauswertungsstelle über die ID zurückgesetzt wurde, erneut in den Datenfluss eingebracht werden. Für das Zurücksetzen eines Dokuments steht kein automatisierter Prozess zur Verfügung. Dieses kann über die E-Mail-Adresse [verfahrensupport@igtig.org](mailto:verfahrensupport@igtig.org) vereinbart werden.

Die Datenflussprotokolle werden durch eine Reduktion der erhaltenen QS-Export-Daten erstellt. Dabei wird das Element `<qs_data>` aus der Datei entfernt. Es verbleiben Header, Protokoll

---

<sup>13</sup> meint auch Anbieter einer Webportal-Software

und Admin-Daten in der Datei. Das übriggebliebene XML enthält innerhalb der ursprünglichen Struktur des Dokuments die bis dahin protokollierten Prüfungen und die sich daraus ergebenden Statusmeldungen der Datensätze und des Dokuments. Der Aufbau des Protokolls ist in der technischen Dokumentation zur Basisspezifikation für Leistungserbringer detailliert beschrieben.

Für das Datenflussprotokoll wird das Attribut `feedback_range` des Elements `<protocol>` auf `dataflow` gesetzt.

Für diesen Verarbeitungsschritt kann das Datenprüfprogramm eingesetzt werden.

Das XML-Protokoll kann von der datenentgegennehmenden Stelle oder von der QS-Software mithilfe einer Template-Definition z.B. nach HTML transformiert werden. Den zuvor beschriebenen Plausibilitätsprüfungen schließen sich auf Bundesebene weitergehende Prüfungen an. Diese führen zu einem weiteren Anwachsen des Prüfprotokolls und der Datenqualität.

Bezogen auf einen bestimmten Datensatz ist es erst nach der letzten abgeschlossenen Prüfung auf Bundesebene möglich, eine Aussage darüber zu treffen, ob sich dieser Datensatz für die Aufnahme in den Bundesdatenpool eignet oder nicht.

Um dem Leistungserbringer das konkrete Ergebnis seiner Datenlieferung in Bezug auf den Bundesdatenpool mitteilen zu können, wird auch das bis zum Schluss weitergeführte Dokument von QS- und patientenidentifizierenden Daten befreit und als Datenflussprotokoll an die DAS versandt, die dann die Aufgabe hat, dieses an die – nur am Pseudonym erkennbaren – Leistungserbringer zu übermitteln.

Die Datenflussprotokolle werden jedoch auch von der Datenannahmestelle für den Fall erstellt, dass eine Weiterleitung der QS-Exporte nicht möglich war. In diesem Fall wird keine Eingangsbestätigung an den Absender verschickt. Um das Dokument als Datenflussprotokoll kenntlich zu machen, muss das Attribut `root/header/protocol/@feedback_range` auf den Wert `dataflow` gesetzt werden.

### Miniprotokoll

Die Erstellung eines vollständigen Datenflussprotokolls durch die DAS ist nur möglich, wenn die ursprüngliche XML-Datei lesbar ist und nach Entfernung der QS-Daten schemakonform bleibt. Andernfalls ist durch die annehmende DAS an dieser Stelle ein reduziertes Datenflussprotokoll („Miniprotokoll“) zu erstellen, das die ID (GUID) des Dokuments (sofern diese zur Verfügung stand und lesbar war, sonst wird diese nicht angegeben) und die konkrete Fehlermeldung enthält. Ein Miniprotokoll ist z.B. zu erstellen, wenn die Exportdatei nicht wohlgeformt ist.



#### Hinweis

Regelungen zu Fehlermeldungen, dem Prüfungsprozess und der Ergebnisprotokollierung, Prüfungsergebnissen, Beziehungen der Vorgangsebene/Dokumentenebene und Rückprotokollen (Bereitstellung eines XSLT für die Transformation) sind der technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

---

## **Regelungen für ein Vorgehen bei Verarbeitungsabbrüchen im besonderen Fehlerfall**

Diese Regelungen dienen dazu, ein abgestimmtes Vorgehen durch alle am Datenfluss beteiligten Stellen zu etablieren, um die Datenbestände in allen Instanzen zu konsolidieren, wenn es zu unerwarteten Störungen in der Verarbeitung von Datenlieferungen in der DAS oder BAS kommt. Solche Störungen sind selten und stellen einen Ausnahmefall dar.

Solche Störungen können in äußerst seltenen Fällen aufgrund einer fehlerhaften automatisierten Verarbeitung von Datenlieferungen in der DAS oder BAS dazu führen, dass fehlerhafte Datenflussprotokolle versandt werden. Dadurch können beim Leistungserbringer unter Umständen mehrere Datenflussprotokolle zu einer GUID eintreffen, die widersprüchliche Einstufungen der Fälle enthalten. Um diesen Umstand an allen am Datenfluss beteiligten Stellen aufzulösen, müssen die involvierten Vorgänge erneut verarbeitet werden.

Das präferierte Vorgehen in dieser Situation ist ein erneuter Versand aller betroffenen Datensätze, die unter der betroffenen GUID gesandt wurden, durch den Leistungserbringer in einer höheren Version. Dieser Versand muss nach Rücksprache mit der Stelle, bei der die fehlerhafte Verarbeitung aufgetreten ist, und unter Einbindung der anderen Stationen im Datenfluss geschehen und wird deshalb telefonisch von der zuständigen DAS beim betroffenen Leistungserbringer initiiert. Für den erneuten Versand ist beim Leistungserbringer ein Export aller betroffenen Vorgangsnummern einschließlich der Erhöhung der zugehörigen Versionsnummer unter einer neuen GUID durchzuführen. Das im Zuge des Exports erstellte XML-Dokument enthält alle Fälle als Update in der jeweils höchsten Versionsnummer, die das Softwaresystem beim Leistungserbringer erstellt, und erhält eine neue und damit unverbrauchte GUID. Somit können alle Datensätze im Datenfluss über alle folgenden Instanzen (DAS, BAS) einschließlich der zugehörigen Rückprotokollierung mittels des Datenflussprotokolls bis hin zum Leistungserbringer regulär verarbeitet werden.

Sollte der Leistungserbringer nicht in der Lage sein, die Daten erneut zu exportieren, so ist der alternative Weg zu wählen, bei dem ein durch alle Instanzen bereits verarbeitetes Dokument (einschließlich Rückprotokollierung mittels Datenflussprotokoll bis hin zum Leistungserbringer) - und somit eine bereits verarbeitete GUID - erneut verarbeitet wird. Es bedarf ebenfalls einer telefonischen Abstimmung aller Stationen im Datenfluss. Auf Leistungserbringerseite muss die Software in der Lage sein, zu einer GUID mehr als ein Datenflussprotokoll aufzunehmen, um die Einstufung der Daten gemäß dem zweiten Datenflussprotokoll analog zu den anderen Verarbeitungsstellen anzupassen.

## B Komponenten

In diesem Kapitel werden die einzelnen Komponenten der Spezifikation beschrieben. Ein Spezifikationspaket bildet die Gesamtheit seiner Spezifikationskomponenten ab. In einem Spezifikationspaket können beispielsweise folgende Komponenten enthalten sein:

- **TechDok** – bezeichnet alle technischen Dokumentationen; diese geben detaillierte Erläuterungen zur Funktionsweise und Verwendung der einzelnen Komponenten. Da es verschiedene spezifisch verfasste TechDoks gibt, wird eine entsprechende Kennzeichnung gleich im Kürzel vermerkt.
  - **TechDok\_EDOK** – Technische Dokumentation für die einrichtungsbezogene QS-Dokumentation
- **QSDOK** – bezeichnet die Access-Datenbank, in der die QS-Dokumentation spezifiziert wird.
- **QSF** – bezeichnet die Access-Datenbank, in der der QS-Filter spezifiziert wird.
- **Schema** – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Sie enthält einzelne XML-Schemata, die festlegen, in welcher Struktur XML-Daten an Schnittstellen im Datenfluss vorliegen müssen.
- **Precheck** – auf der Komponentenebene ist dies eine ZIP-Datei, die Schemata für alle administrativen Daten enthält, die ab dem Export dieser Spezifikation gültig sind. Diese Schemata sind ausschließlich für die Verarbeitung der Datenservices der DAS und BAS relevant.
- **Ausfuellhinweise** – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Sie enthält einzelne HTML-Dateien für jedes Modul, die mit den Kürzeln der einzelnen Module benannt sind.
- **Anwenderinformationen** – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Die ZIP-Datei enthält einzelne HTML-Dateien für jedes im QS-Filter definiertes Modul, die mit den Kürzeln der einzelnen Modulauslöser benannt sind.
- **Dokuboegen** – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Sie beinhaltet die Dokumentationsbögen als einzelne PDF-Dateien für jedes Modul, die mit den Kürzeln der einzelnen Module benannt sind. Die Dokumentationsbögen bilden einige wichtige Datenbankinhalte<sup>14</sup> ab.
- **UebersichtAenderungen** – ist eine PDF-Datei mit der Übersicht über die Änderungen in der Spezifikation zu allen Vorversionen des Erfassungsjahres und zur letzten Version des Vorjahres. Hier werden die Änderungen übersichtlich zusammengefasst und erläutert.
- **Protocol** – bezeichnet das Stylesheet zur Darstellung des Rückprotokolls (XSLT).
- **Komponentenliste** – bezeichnet die Auflistung aller im Spezifikationspaket enthaltenen Komponenten mit Bezug zu Versionsnummer und Veröffentlichungsdatum (CSV).

---

<sup>14</sup> Die Papierform ist hier nur als eine Abbildung des Eingabeformulars oder der Eingabemaske zu verstehen. Verbindlich sind daher nur die Inhalte der Datenbank zur QS-Dokumentation.

Hilfsprogramme werden ggf. ebenfalls als Komponenten in ein Spezifikationspaket aufgenommen. Hierbei kann es sich beispielsweise um **Verschlüsselungsprogramme**, ein **Datenprüfprogramm** oder ein **PSP** (Pseudonymisierungsprogramm) handeln. Die Angaben der Betriebsart und des Exportformats erfolgen nur dann, wenn sich die Komponenten durch diese Merkmale unterscheiden.

**V<Versionsnummer>**: Die Versionierung erfolgt in ganzen Zahlen, die zweistellig angegeben sind (unter 10 mit einer vorstehenden 0, z.B. V01).

Weitere mögliche Spezifikationskomponenten sind der Technischen Dokumentation zur Basispezifikation für Leistungserbringer zu entnehmen.

### **Spezifikationsdatenbanken**

Als Access-Datenbanken zur Verfügung gestellte Spezifikationskomponenten dienen der (automatisierten) Erstellung von z.B. Software für QS-Filter, QS-Dokumentation. Folgende Spezifikationskomponenten werden als Access-Datenbanken (MS Access 2007-2013) zur Verfügung gestellt:

#### **Datenbank zur QS-Dokumentation (QSDOK)**

Die Datenbank zur QS-Dokumentation dient der Spezifikation von Datenerhebung und -erfassung unter Berücksichtigung von Plausibilitätsprüfungen und zu exportierenden Datenfeldern. Einige wichtige Datenbankinhalte werden über die Dokumentationsbögen (Dokubögen) abgebildet.

Folgende Spezifikationskomponente ergänzt die Datenbank um Hinweise und Informationen:

- Die Ausfüllhinweise (Ausfuellhinweise) dienen als Hilfestellung bei der Dokumentation durch den Anwender. Die Namen der HTML-Dateien für einzelne Datenfelder sind in der Datenbank (BogenFeld.ahinweis) hinterlegt.

Die Datenbank zur QS-Dokumentation ist in Kapitel B 2 erläutert.

#### **Datenbank zum QS-Filter (QSF)**

Die Datenbank zur QS-Dokumentation dient der Spezifikation zur Auslösung von Dokumentationsmodulen. Die Auslösebedingungen pro Modul werden als Übersicht in Form der Anwenderinformationen (Anwenderinformationen) zur Verfügung gestellt.

Die Datenbank zum QS-Filter ist in Kapitel B 1 erläutert.

### **Tabellenstruktur der Datenbanken**

Die Tabellen und deren Spalten (Attribute) unterliegen einem einheitlichen Namensschema. Erlaubte Zeichen sind die Buchstaben a–z, A–Z und die Ziffern 0–9. Umlaute und Sonderzeichen werden nicht verwendet. Das erste Zeichen eines Namens darf keine Ziffer sein.

Ein Tabellename beginnt immer mit einem Großbuchstaben und ein Attributname mit einem Kleinbuchstaben. Wenn ein Name aus mehreren Teilen (z. B. Substantiven) besteht, so beginnt jeder nachfolgende Namensteil mit einem Großbuchstaben.

BasisTyp (Tabelle)  
idBasisTyp (Spalte)

Für jede Tabelle ist in der Spezifikation ein Primärschlüssel definiert, der nach folgendem Schema aufgebaut ist:

id<TabellenName>

Der Ausdruck in spitzen Klammern ist ein Platzhalter für den Namen der Tabelle. Die meisten Tabellen haben einen einfachen Primärschlüssel vom Typ `AUTOINCREMENT`. Zusätzlich enthalten derartige Tabellen mindestens ein identifizierendes Attribut<sup>15</sup>, welches durch Setzen eines weiteren, eindeutigen Indexes (bestehend aus einem oder mehreren Attributen) definiert ist.

---

### Beispiele:

Identifizierendes Attribut: Attribut `name` in Tabelle `BasisTyp`

Identifizierende Attributkombination: Attribute `code` und `fkSchlüssel` in Tabelle `SchlüsselWert`

---

Es gibt auch Tabellen, deren einziger eindeutiger Schlüssel der Primärschlüssel ist. Ein Beispiel ist die Tabelle `MussKann` mit dem Primärschlüssel `idMussKann` vom Typ `TEXT(1)` (entspricht `VARCHAR(1)`). Diese Tabellen sind als einfache „Nachschlagtabellen“ zu interpretieren. Im Fall der Tabelle `MussKann` soll im entsprechenden Fremdschlüsselfeld der verknüpften Detailtabelle durch das Datenbankschema gewährleistet werden, dass nur ein `M` oder `K` eingegeben werden darf.

Die Namen von Fremdschlüsseln sind analog zum Namen der Primärschlüssel aufgebaut:

fk<FremdTabellenName>

Die Namensgebung von Primär- und Fremdschlüsseln vereinfacht den Aufbau von komplexeren Abfragen, welche sich über mehrere Entitäten erstrecken (Inklusionsverknüpfungen, Joins).

Die Fremdschlüsselattribute (Namen beginnen mit `fk`) wurden als Datenbankattribute zum Nachschlagen eingerichtet. Zum Beispiel wird beim Fremdschlüsselattribut `fkModul` in der Tabelle `Tds` nicht mehr der Primärschlüssel des jeweiligen Moduls, sondern der Name des Moduls angezeigt.



### Hinweis

Diese Änderung betrifft nur die Anzeige, nicht jedoch die Struktur der Datenbank.

---

Sind zwei Tabellen mehrfach durch Schlüssel-Fremdschlüssel-Beziehungen miteinander verknüpft, so kann der Name eines Fremdschlüssels auch folgendermaßen aufgebaut sein:

fk<FremdTabellenName><Rolle>

---

<sup>15</sup> Oder sie enthalten eine identifizierende Attributkombination, die einen eindeutigen Schlüssel definiert.

<Rolle> ist der Platzhalter für eine zusätzliche Qualifizierung der Relation.

N-M-Beziehungen werden wie üblich über Verknüpfungstabellen realisiert. In der Spezifikation haben Verknüpfungstabellen gewöhnlich keinen Primärschlüssel<sup>16</sup>, jedoch einen eindeutigen Schlüssel, der über die Fremdschlüsselfelder definiert ist.

Folgende Attribute treten in vielen Tabellen auf:

- `name` ist in der Regel als technischer Name zu verstehen. Zum Beispiel wird `Feld.name` als Variablenname in den Plausibilitätsregeln verwendet.
- `bezeichnung` ist eine kurze Beschreibung. Zum Beispiel ist `TdsFeld.bezeichnung` der Text, welcher ein Feld auf einem Eingabeformular beschreibt.
- `bedingung` enthält einen logischen Ausdruck. Prominentester Vertreter dieses Attributtyps ist das Attribut `bedingung` in der Tabelle `ModulAusloeser`.

---

<sup>16</sup> Hier: Primärschlüssel im Sinne der Access-Definition eines Primärschlüssels. Streng genommen wird über die beiden Fremdschlüssel ein neuer Primärschlüssel definiert.

# 1 QS-Filter

Der QS-Filter definiert, unter welchen Bedingungen ein Modul ausgelöst wird; in der Datenbank zum QS-Filter und den dazugehörigen Anwenderinformationen sind die Informationen hierzu hinterlegt. Die Auslösekriterien werden für jeden Fall geprüft. Sind die Kriterien für mindestens einen Fall erfüllt, gilt das jeweilige Modul der externen Qualitätssicherung (NWIES und/oder NWIEA) für eine Einrichtung als dokumentationspflichtig.

## 1.1 Anmerkungen zur Struktur der Spezifikationsdatenbank für QS-Filter

Die QS-Filter-Spezifikation ist in einer relationalen Datenbank abgelegt. Zurzeit wird sie ausschließlich als Access-Datenbank (MS Access 2007-2013) zur Verfügung gestellt. Der Name der QS-Filter-Spezifikation richtet sich nach folgendem Schema:

<Erfassungsjahr>\_QESUE\_QSF\_V<Versionsnummer>.mdb

<Versionsnummer> bezeichnet die 2-stellige Versionsnummer (z.B. 01).

---

### Beispiel:

Im Erfassungsjahr 2017 ist die QS-Filter-Spezifikation 2017\_QESUE\_QSF\_V01.mdb<sup>17</sup> gültig. Die Kennung 2017 gilt für das Erfassungsjahr 2017.

---

Weiterführende Erläuterungen zum Benennungsschema für Spezifikationskomponenten sind der Einleitung in Abschnitt 1.1.2 zu entnehmen.

Die Tabellenstruktur der Spezifikationsdatenbank wird in Abschnitt B Komponenten beschrieben.

## 1.2 Grundlegende Tabellen der Datenbank

Dieser Abschnitt beinhaltet die Darstellung der grundlegenden Tabellen der Spezifikationsdatenbank mit ihren zugehörigen Datensätzen ausgehend von den jeweiligen Modulen.

### 1.2.1 Module (Datensätze der QS-Dokumentation)

In der Tabelle `Modul` sind Referenzen auf die Module hinterlegt, deren Dokumentationspflicht durch den QS-Filter ausgelöst werden kann.

---

<sup>17</sup> Die Versionsnummer der gültigen Spezifikation (z.B. V01, V02, usw.) ist dem zuletzt veröffentlichten Update zu entnehmen.

Hinter jedem Modul verbirgt sich ein Datensatz der korrespondierenden Spezifikation für die QS-Dokumentation. Diese Datensätze bilden in der Spezifikation für QS-Dokumentation den auszulösenden Dokumentationsbogen (z.B. der Software oder des Webportals<sup>18</sup>) ab.

---

### Beispiele:

Einrichtungen, die ambulante und stationäre Fälle am Krankenhaus behandeln

- Der Modulauslöser `NWIES_LKG` löst den QS-Datensatz `NWIES` aus.
- Der Modulauslöser `NWIEA_LKG` löst den QS-Datensatz `NWIEA` aus.

Einrichtungen, die ambulante vertragsärztliche oder stationäre belegärztliche Leistungen erbringen

- Der Modulauslöser `NWIES_KV` löst den QS-Datensatz `NWIES` aus.
  - Der Modulauslöser `NWIEA_KV` löst den QS-Datensatz `NWIEA` aus.
- 

Die Module der Spezifikation für QS-Dokumentationssoftware sind in der gleichnamigen Tabelle definiert. Eine Zuordnung ist über das Attribut `Modul.name` möglich. Die Datensätze der QS-Dokumentation werden in Abschnitt B 2.2.1 beschrieben.

Die Struktur der Tabelle `Modul` wird im Folgenden beschrieben.

Tabelle 6: Struktur der Tabelle `Modul`

Feldname	Datentyp	Bemerkung
<code>idModul</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name des Moduls (Identifizierendes Attribut)
<code>bezeichnung</code>	TEXT	Beschreibender Text für den QS-Datensatz
<code>fkSchluesselWert</code>	INTEGER	Verweis auf denjenigen Kode des Schlüssels <code>Modul</code> (Tabelle <code>SchluesselWert</code> ), welcher dem betreffenden Datensatz zugeordnet ist.
<code>fkModulParent</code>	INTEGER	Verweis auf das Eltern-Modul eines Zählleistungsbereichs Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
<code>zaehlLb</code>	BOOLEAN	Wenn WAHR, existiert kein entsprechender QS-Datensatz, es handelt sich um einen „Zählleistungsbereich“ zur separaten Darstellung in der Sollstatistik. Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.

---

<sup>18</sup> Die Kassenärztlichen Vereinigungen als DAS stellen den Vertragsärzten in ihrem Zuständigkeitsbereich Anwendungen zur Erfassung und Übertragung von Daten der einrichtungsbezogenen QS-Dokumentation, z.B. webbasiert, zur Verfügung (ggf. mit Unterstützung durch die Kassenärztliche Bundesvereinigung). Im Krankenhaussektor wird von Softwareherstellern eine QS-Filtersoftware umgesetzt.

Feldname	Datentyp	Bemerkung
ausloeseModul	BOOLEAN	Wenn WAHR, existiert ein entsprechender QS-Datensatz, der ausgelöst werden kann. Dieses Modul kann gleich dem Exportmodul sein, z.B. NWIES.
exportModul	BOOLEAN	Wenn WAHR, handelt es sich um ein Exportmodul, z. B. NWIES_LKG. Das Exportmodul kann ungleich dem im Modulauslöser referenzierten Modul sein, z. B. NWIES.
direkt	BOOLEAN	handelt es sich um ein direktes Datenexportverfahren? Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
indirekt	BOOLEAN	handelt es sich um ein indirektes Datenexportverfahren?
pid	BOOLEAN	handelt es sich um ein Modul zur Follow-up-Erhebung? <sup>19</sup> Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
qskh	BOOLEAN	handelt es sich um ein Modul der QSKH-Richtlinie? Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
qesue	BOOLEAN	handelt es sich um ein Modul der Qesü-Richtlinie?

### 1.2.2 Struktur der Datensatzdefinitionen

Die Module werden über den definierten Modulauslöser als dokumentationspflichtig erkannt, indem die hinterlegte Bedingung mit den im KIS oder in den Abrechnungsdaten der Kassenärztlichen Vereinigungen gespeicherten Daten geprüft wird. Ist die Bedingung erfüllt, wird das Modul ausgelöst. Da die in der Bedingung enthaltenen Felder im KIS oder in den Abrechnungsdaten der Kassenärztlichen Vereinigungen vorliegen müssen, ist der Eingangsdatsatz gemäß § 301 SGB V und § 295 SGB V in der Spezifikationsdatenbank hinterlegt. Die definierten Felder der Datensätze und der Filterbedingungen sind analog zur Spezifikation für die QS-Dokumentation aufgebaut.

In den nachfolgenden Abschnitten dieses Unterkapitels wird die grundlegende Struktur der Tabellen der Datensatzdefinitionen beschrieben und die Definitionen der Datenfelder erläutert.

#### Datensätze

Jeder Datensatz besteht aus Teildatensätzen, welche ausgehend von einem Basisteildatensatz hierarchisch angeordnet sind. Folgende Datensätze sind definiert:

<sup>19</sup> Gemeint ist eine Follow-up-Erhebung mittels PID-Verfahren.

- QS-Filter-Datensätze

Für jeden dieser Datensätze wird in der Tabelle Ds ein Eintrag angelegt.

Tabelle 7: Struktur der Tabelle Ds

Feldname	Datentyp	Bemerkung
idDs	INTEGER	Primärschlüssel
name	TEXT	Technischer Name des Datensatzes (Identifizierendes Attribut)
bezeichnung	TEXT	Beschreibender Text

### Teildatensätze

Die Definition von Teildatensätzen befindet sich in der Tabelle Tds der Datenbank (Tabelle 8). Jeder Teildatensatz ist eindeutig durch seinen Namen (z.B. FALL, PROZ oder DIAG) charakterisiert.

Tabelle 8: Struktur der Tabelle Tds

Feldname	Datentyp	Bemerkung
idTds	INTEGER	Primärschlüssel
name	TEXT	Technischer Name des Teildatensatzes (Identifizierendes Attribut)
bezeichnung	TEXT	Beschreibender Text
fkTds	INTEGER	Optionaler Fremdschlüssel zu einem Mutterteildatensatz
fkDs	INTEGER	Bezug des Teildatensatzes zum übergeordneten Datensatz in der Tabelle Ds, z.B. QS-Filter-Datensatz oder QS-Filter-Datensatz für Sollstatistik
fkRelation-Typ	TEXT(1)	Relationstyp, bezieht sich auf die Relation zum Mutterteildatensatz: * Eine beliebige Anzahl von Teildatensätzen darf angelegt werden! ? Höchstens ein Teildatensatz darf angelegt werden! + Mindestens ein Teildatensatz muss angelegt werden! 1 Genau ein Teildatensatz muss angelegt werden!
fkEindeutigTdsFeld	INTEGER	Fremdschlüssel auf ein TdsFeld, das mehrfach vorhandene Teildatensätze eines Datensatzes identifiziert
sortierNr	INTEGER	Angabe zur Reihenfolge der Teildatensätze

In der Tabelle `Tds` ist eine Hierarchie der Teildatensätze definiert. Der Ausgangspunkt („root“) für die Teildatensatzhierarchie eines Datensatzes (z.B. QS-Filter-Datensatz) ist immer der Basisteildatensatz (daraus folgt: Teildatensatz<sup>20</sup> `fkTds = NULL` in der Tabelle `Tds`). Über die in den restlichen Teildatensätzen des Moduls definierten Bezüge zu den Mutterteildatensätzen und Relationstypen lässt sich ein Hierarchiebaum der Teildatensätze aufbauen.

Jeder Datensatz besteht aus

- genau einem Basisteildatensatz
- ggf. einem oder mehreren weiteren Teildatensätzen (= Kindteildatensätze)

---

### Beispiele:

Der QS-Filter-Datensatz `DATENSATZ_301` besitzt den Basisteildatensatz `FALL` und die sechs Kindteildatensätze `DIAG`, `PROZ` und `ENTGELT`

---

### Felder der Teildatensätze

Die Felder eines Teildatensatzes sind in der Tabelle `TdsField` definiert. Jedes Feld eines Teildatensatzes (kurz TDS-Feld) ist eindeutig charakterisiert durch die Zugehörigkeit zu einem Teildatensatz und zum referenzierten Feld. Jedes Feld darf also nur einmal in einem Teildatensatz verwendet werden. Listenfelder erfordern einen Wert `> 1` beim Attribut `elemente`.

### Felder

Ein Feld wird eindeutig über seinen technischen Namen definiert. Jedes Feld hat einen Basistyp (z.B. `SCHLUESSEL`, `NUMSCHLUESSEL`, `ZAHL`). Schlüsselfelder erfordern zusätzlich einen Schlüssel (siehe Tabelle 9).

### Basistypen

Das Hauptmerkmal eines Basistyps ist der technische Typ eines Eingabefeldes (z.B. Zeichenkette, numerischer Typ, Datum usw.). Weiteres Charakteristikum ist die Beschreibung des Eingabeformats. Die Basistypen sind Voraussetzung für die Beschreibung einer formalen Regelsyntax. Das identifizierende Merkmal eines Basistyps ist sein technischer Name (Attribut `name`).

Tabelle 9: Struktur der Tabelle `BasisTyp`

Feldname	Datentyp	Bemerkung
<code>idBasisTyp</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name (muss eindeutig sein)
<code>bezeichnung</code>	TEXT	Beschreibender Text
<code>formatAnweisung</code>	TEXT	Regulärer Ausdruck für die Formatprüfung

---

<sup>20</sup> Es darf nur eine Definition eines Basisteildatensatzes existieren.



### Hinweis

In Zeichenketten (Basistyp `TEXT`) sind alle Zeichen des ASCII-Formats mit einem Kode  $\geq 32$  erlaubt. Ausgenommen sind das Semikolon, die doppelten Anführungsstriche und Hochkommata.

- Es gibt zwei Arten von Schlüsseln: numerische und nichtnumerische.
- Das Komma trennt die Nachkommastellen, Vorzeichen + und – sind erlaubt.
- Das Datumstrennzeichen ist der Punkt.

### Schlüssel

Identifizierendes Merkmal eines Schlüssels ist sein technischer Name. Die meisten Schlüsselcodes sind in der Tabelle `SchluesseleWert` (Tabelle 10) definiert. Externe Schlüsselkataloge (z.B. OPS, EBM oder ICD10) sind von den entsprechenden Anbietern zu beziehen.

Tabelle 10: Struktur der Tabelle `Schluessele`

Feldname	Datentyp	Bemerkung
<code>idSchluessele</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name (muss eindeutig sein)
<code>bezeichnung</code>	TEXT	Beschreibender Text
<code>extern</code>	BOOLEAN	Zeigt an, ob der Schlüssel in der Tabelle <code>Schluessele</code> oder in einer externen Tabelle gespeichert ist.
<code>externVerweis</code>	TEXT	Verweis auf externe Quelle des Schlüsselkataloges
<code>zahl</code>	BOOLEAN	Wenn WAHR, wird das Attribut <code>code</code> der zugehörigen Schlüsselwerte als ganze Zahl interpretiert, ansonsten als Zeichenkette.

Schlüsselcodes können auf zwei Arten kodiert werden. Wenn das Attribut `zahl = WAHR`, so werden die Codes als ganze Zahl interpretiert. Ansonsten werden sie als Zeichenketten angesehen. In der Syntax der Auslösebedingungen werden die letztgenannten Codes in einfache Hochkommata gesetzt.

### Beispiel:

Attribut `zahl` bei Schlüsselfeldern

Felder des Basistyps `NUMSCHLUESSEL` haben das Attribut `zahl = TRUE`.

Felder des Basistyps `SCHLUESSEL` haben das Attribut `zahl = FALSE`. Es handelt sich um alphanumerische Schlüssel, die Buchstaben, Ziffern oder Sonderzeichen verwenden (z. B. `ypN0`). Hierbei kann es sich auch um Werte handeln, die lediglich Ziffern verwenden, aber mit führender Null beginnen (z.B. `01`).

## Externe Schlüsselkataloge

Externe Schlüsselkataloge sind über das Attribut `extern` deklariert. Externe Schlüsselkataloge werden nicht vom IQTIG bereitgestellt und daher auch nicht verantwortet.

Hinweise zu den Bezugsquellen sind in der Spalte `externVerweis` zu finden (z.B. <http://www.dimdi.de>). Ein Verweis auf eine Bezugsquelle kann unabhängig vom Attribut `extern` angegeben werden (siehe Schlüssel JN).



### Hinweis

Das Feld `PERSONENKREIS` referenziert zwei externe Quellen, da dieses Feld sowohl die Schlüsselwerte nach § 301 als auch nach § 295 (KVD-Datensatzbeschreibung) abbildet. Es gelten die folgenden externen Verweise:

- besonderer Personenkreis gemäß Anlage 2 zur Vereinbarung nach § 301 Abs. 3 SGB V (Syntaxvariable `DATENSATZ_301 : FALL : PERSONENKREIS`)  
externer Verweis: § 301-Vereinbarung: <http://www.dkgev.de>,
- besondere Personengruppe gemäß der Datensatzbeschreibung KVD (Kassenärztliche Vereinigung-Datentransfer) (Syntaxvariable `DATENSATZ_295 : FALL : PERSONENKREIS`)  
externer Verweis: Datensatzbeschreibung KVD: <http://www.kbv.de>



### Achtung

Der Softwareanbieter bzw. die Kassenärztliche Vereinigung hat dafür Sorge zu tragen, dass die aktuellen externen Schlüsselkataloge in der Software verwendet werden.

## Schlüsselwerte

Identifizierendes Merkmal ist hier eine Kombination der Spalten `fkSchluessel` und `code`. Das bedeutet, dass jeder Schlüsselcode innerhalb eines Schlüssels nur einmal vorkommen darf.

Tabelle 11: Struktur der Tabelle `SchlüsselWert`

Feldname	Datentyp	Bemerkung
<code>idSchluesselWert</code>	INTEGER	Primärschlüssel
<code>fkSchluessel</code>	INTEGER	Fremdschlüssel zur Tabelle <code>Schlüssel</code>
<code>code</code>	INTEGER	0, 1, 2 ...
<code>bezeichnung</code>	TEXT	Textliche Definition des Schlüsselwertes

**Hinweis**

Die Schlüsselwerte lassen sich am einfachsten über die Abfrage `Schlüsselkodes` ermitteln.

---

**1.2.3 Funktionen**

Eine Funktion ist gekennzeichnet durch ihren Namen, an den sich unmittelbar (ohne Leerzeichen) ein Listenausdruck anschließt. Funktionen ohne Übergabeparameter werden ähnlich wie in C oder Java durch ein Klammerpaar abgeschlossen. Funktionen können nicht nur in Regeln, sondern auch zur Berechnung von Exportfeldern genutzt werden (Abschnitt B 2.4.3).

Der aktuelle Stand der in der Syntax verwendeten Funktionen ist in der Tabelle `SyntaxFunktion` der Spezifikation zu finden. In der QSDOK-Spezifikationsdatenbank der einrichtungsbezogenen QS-Dokumentation sind derzeit keine Funktionen hinterlegt. In der QSF-Spezifikationsdatenbank der einrichtungsbezogenen QS-Dokumentation finden verschiedene Funktionen Anwendung.

In den nachfolgenden Beispielen gilt folgende Notation für Funktionen:

```
<BASISTYP> <FUNKTIONSNAME>([ <BASISTYP> <VARNAME>{ ;
<BASISTYP> <VARNAME>}])
```

mit

- { }                   Wiederholung
  - [ ]                   Option
  - <BASISTYP>       Basistyp der Variablen
  - <VARNAME>       Name der Variablen
- 

**Beispiele<sup>21</sup>:**

```
DATUM aktuellesDatum( )
```

Funktion ohne Übergabeparameter und mit Ergebnistyp DATUM

```
DATUM Minimum(DATUM DATUMLISTE)
```

Funktion mit Ergebnis vom Typ DATUM, die das Minimum einer Liste von Datumsangaben (DATUMLISTE) liefert.

```
JAHRDATUM jahreswert(DATUM EINDATUM)
```

Funktion mit Ergebnis vom Typ JAHRDATUM

---

<sup>21</sup> Die im Beispiel genannten Felder sind NICHT Bestandteil der Spezifikationsdatenbank für den QS-Filter und dienen an dieser Stelle lediglich der Erläuterung.

Es kommen auch verschachtelte Funktionsaufrufe (z.B. funktionA(funktionB())) oder arithmetische Ausdrücke als Funktionsargumente (z.B. funktion(x+y)) vor. Häufig wird nur die Signatur von Funktionen bereitgestellt.

### Hinweise für die Implementierung von Funktionen

Als Hilfestellung für die Ausprogrammierung wird bei manchen Funktionen ein Pseudocode bereitgestellt. Der Pseudocode ergänzt die Syntax der Plausibilitätsregeln um folgende Sprachelemente:<sup>22</sup>

- Befehlszeilen werden mit Semikolon abgeschlossen ;
- Wertzuweisungen mit dem Operator :=

```
A := B + C;
```

- Auswahanweisungen

```
if (<Bedingung>){
    ...
}
else {
    ...
}
```

Hinter <Bedingung> verbirgt sich ein logischer Ausdruck, der der Syntax der Plausibilitätsregeln gehorcht.

- Blöcke werden durch geschweifte Klammern definiert.

```
{
...
}
```

- Innerhalb einer Funktion sind die Argumentvariablen verfügbar.
- Eine Variable, die den gleichen Namen wie die Funktion hat, muss am Ende mit return zurückgegeben werden.

#### 1.2.4 OPS-Listen

Die Struktur der Tabellen `OPSListe` und `OPSWert` entsprechen den gleichnamigen Tabellen der Spezifikationsdatenbank für die QS-Dokumentation. Aufgrund des Einrichtungsbezugs sind für das Verfahren QS WI nur die entsprechenden Tabellen in der Spezifikationsdatenbank für den QS-Filter relevant.

#### 1.2.5 ICD-Listen

Die Struktur der Tabellen `ICDListe` und `ICDWert` entsprechen den gleichnamigen Tabellen der Spezifikationsdatenbank für die QS-Dokumentation. Aufgrund des Einrichtungsbezugs sind

---

<sup>22</sup> Der Pseudocode erhebt nicht den Anspruch auf formale Korrektheit.

für das Verfahren QS WI nur die entsprechenden Tabellen in der Spezifikationsdatenbank für den QS-Filter relevant.

### Umgang mit Kennzeichen zur Diagnosesicherheit

Das Zusatzkennzeichen zur Diagnosesicherheit ist in den ICD-Listen der Spezifikationsdatenbanken nicht enthalten.

ICD-Kodes mit dem Zusatzkennzeichen A (Ausschluss) sind bei der Auslösung nicht zu berücksichtigen. Bei der Auslösung von ambulanten Fällen ist die Auslösung nur mithilfe von gesicherten Einschlussdiagnosen zulässig.



#### **Achtung Auslösung**

Bei einer prozessbegleitenden Auslösung über eine QS-Filtersoftware ist sicherzustellen, dass Aktualisierungen im KIS zwischen der Anwendungssoftware und der QS-Dokumentationssoftware kommuniziert werden. Grundsätzlich ist hier eine Übereinstimmung aller Diagnosen mit der Entlassungsanzeige des Behandlungsfalles sicherzustellen.

ICD-Kodes mit dem Zusatzkennzeichen A (Ausschluss) sind bei der Prüfung auf Ausschlusslisten nicht zu berücksichtigen. Im QS-Filter-Algorithmus wird die Syntaxvariable `DIAG` daher daraufhin geprüft, ob es sich um eine gesicherte Diagnose handelt. Hierfür wird die folgende Teilbedingung verwendet:

```
format (DIAG; '[a-zA-Z][0-9]{2}(\.[0-9]{1,2})?([#\+*\+!])?([Gg]){1}([RLBr1b])?') = WAHR
```

### 1.2.6 FAB-Listen

Jede FAB-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `FABListe`), welcher nach folgendem Schema gebildet wird:

```
{<TEXT>}FAB{<TEXT>}
```

Hinter `<TEXT>` verbirgt sich ein frei wählbarer Name (Erlaubte Zeichen: A–Z, a–z, 0–9, Umlaute sind nicht erlaubt). Die `{}`-Ausdrücke sind optional.



#### **Hinweis zu NWIES\_KV und NWIEA\_KV Teilbedingung TeilStr(LANR;8;2) IN FAB\_NWI\_LANR**

Sind die notwendigen Informationen zu den Facharztgruppen der Tabelle `FAB_NWI_LANR` im jeweiligen Abrechnungssystem separat abgelegt, können diese von den Kassenärztlichen Vereinigungen anstelle der `LANR` zur Prüfung herangezogen werden.

Die Prüfung muss hierbei jederzeit zum selben Ergebnis führen wie die Prüfung der letzten beiden Stellen der `LANR`.

### 1.2.7 EBM-Listen

Jede EBM-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `EBMListe`), welcher nach folgendem Schema gebildet wird:

$$\{ \langle \text{TEXT} \rangle \_ \} \text{EBM} \{ \_ \langle \text{TEXT} \rangle \}$$

Hinter `<TEXT>` verbirgt sich ein frei wählbarer Name (Erlaubte Zeichen: A–Z, a–z, 0–9, Umlaute sind nicht erlaubt). Die `{}`-Ausdrücke sind optional.

#### String-Vergleich bei EBM-Kodes

EBM-Ziffern können auf Landesebene um weitere Zeichen ergänzt werden. Das Feld `Gebührenordnungsziffer` gemäß EBM-Katalog `[EBM]` hat daher die Feldlänge 7.

In der Spezifikationsdatenbank sind die EBM-Kodes in Tabelle `EBMWert` fünfstellig – ohne ergänzenden Zeichen – definiert. Für die Prüfung, ob zwei Kodes identisch sind, genügt kein einfacher Stringvergleich. Stattdessen wird ein Stringvergleich der Normcodes<sup>23</sup> durchgeführt, um die Übereinstimmung zwischen dem dokumentierten Kode und dem einer EBM-Liste zu ermitteln.

### 1.2.8 Entgelt-Listen

Jede Entgelt-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `EntgeltListe`), welcher nach folgendem Schema gebildet wird:

$$\langle \text{Fallart} \rangle \{ \_ \langle \text{TEXT} \rangle \} \text{ENTGELT} \{ \_ \langle \text{TEXT} \rangle \}$$

Hinter `<TEXT>` verbirgt sich ein frei wählbarer Name (Erlaubte Zeichen: A–Z, a–z, 0–9, Umlaute sind nicht erlaubt). Die `{}`-Ausdrücke sind optional.

---

#### Beispiel:

`STAT_BELEG_ENTGELT`      belegärztliche Fälle (`FALLART` = stationär; gemäß `[ENTGELTARTSTAT]`)

---

### 1.2.9 Versionsverwaltung

Jede Spezifikationsdatenbank hat eine Version. Die Versionsinformation ist in der Tabelle `Version` der Eintrag, welcher den Attributwert `gueltig` = WAHR besitzt.

Die wichtigsten Eigenschaften einer Version sind der Versionsname (Attribut `name`) und der Gültigkeitszeitraum (Attribute `ab` und `bis`).



#### Achtung

Der Gültigkeitszeitraum einer Version ist in der fallbezogenen QS-Basispezifikation für Leistungserbringer in der Regel ein Erfassungsjahr (z.B. Aufnahme zwischen dem

---

<sup>23</sup> Jeder EBM-Kode lässt sich entweder als Kode mit ergänzenden Zeichen (Normcode + ergänzende Zeichen) oder als Kode ohne ergänzende Zeichen (Normcode) darstellen.

01.01.2017 und dem 31.12.2017). Die Auslösung der einrichtungsbezogenen QS-Dokumentation bezieht sich bei den Modulen NWIEA und NWIES nur auf **die ersten beiden Quartale des Erfassungsjahres**.

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation ist beispielsweise für das Erfassungsjahr 2017 vom 01.01.2017 bis zum 30.06.2017 gültig. Die Prozesse des Verfahrens können außerhalb dieses Zeitraums liegen.

Versionen können den Status `in Entwicklung` oder `final` haben. Diese Zustände werden in der Nachschlagetabelle `VersStatus` verwaltet. Das Attribut `gueltig` zeigt die gültige Version der Datenbank an. Nur eine einzige Version darf als gültig markiert sein. Darüber hinaus verwaltet die Tabelle `Version` die Historie der Versionen<sup>24</sup>: Welche Vorgängerversion vorher gültig war, kann über das Attribut `fkVersion` ermittelt werden.

Tabelle 12: Struktur der Tabelle `Version`

Feldname	Datentyp	Bemerkung
<code>idVersion</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name der Version (Identifizierendes Attribut)
<code>bezeichnung</code>	TEXT	Beschreibender Text
<code>ab</code>	DATUM	Anfang des Gültigkeitszeitraumes
<code>bis</code>	DATUM	Ende des Gültigkeitszeitraumes
<code>pub</code>	DATUM	Datum der Publikation
<code>gueltig</code>	BOOLEAN	gültige Version (nur ein Eintrag darf als gültig markiert sein)
<code>fkVersion</code>	INTEGER	Bezug zur Vorgängerversion
<code>fkVersStatus</code>	CHAR	Bezug zum Status einer Version (Tabelle <code>VersStatus</code> ): E = in Entwicklung F = finale Version U = Update der finalen Spezifikation

Das Attribut `Modul.fkVersion` wird verwendet, um die aktuelle Version eines Moduls kenntlich zu machen.

### 1.2.10 Meta-Tabellen

In den Tabellen `TabellenStruktur` und `TabellenFeldStruktur` werden die Tabellen und ihre Attribute aufgelistet. Der Inhalt wird automatisch generiert (Abschnitt B 2.5.2).

<sup>24</sup> Die Inhalte der Vorversionen sind nicht Teil der aktuellen Spezifikationsdatenbank.

### 1.2.11 DB-Änderungen gegenüber der Vorgängerversion

Die Tabellen `DeltaNeu`, `DeltaGeloescht` und `DeltaAttribut` zeigen die Änderungen zur letzten gültigen Spezifikation des Vorjahres und zur Vorversion an (Abschnitt B 2.5.2).

## 1.3 Der QS-Filter-Datensatz

Verbindlich für ein Erfassungsjahr ist die Datenfeldbeschreibung, welche in der gültigen Spezifikationsdatenbank definiert ist. Hier werden die Datensätze `DATENSATZ_301` und `DATENSATZ_295` dargestellt. Beide bilden jeweils den entsprechenden QS-Filter-Eingangsdatensatz ab (Tabelle 13).

Tabelle 13: Ausschnitt der Tabelle `Ds`

idDs	name	bezeichnung
1	DATEN-SATZ_301	Datenübermittlung nach § 301 Abs. 3 SGB V, ergänzt um die Festlegungen der Vereinbarung nach § 120 Abs. 3 SGB V
3	DATEN-SATZ_295	Datensatzbeschreibung KVDT [KBV_ITA_VGEX_Datensatzbeschreibung_KVDT], Kassenärztliche Bundesvereinigung



#### Achtung

#### Datenverlust durch Fehlbezug und falsche Falleinschränkung

Die Bezeichnungen der Datensätze sowie deren Felder nehmen Bezug auf § 301 SGB V und § 295 SGB V. Diese Gesetzesgrundlagen beziehen sich lediglich auf den Kontext der Gesetzlichen Krankenversicherung. Der QS-Filter-Datensatz hingegen bezieht sich prinzipiell auf alle behandelten Patienten/Fälle.

Da in das Qesü-Verfahren QS WI nur gesetzlich versicherte Patienten einbezogen werden, kann eine Dokumentationspflicht nur identifiziert werden, wenn das Institutionskennzeichen der Krankenkasse der Versichertenkarte mit der Zeichenkette '10' beginnt.

### 1.3.1 Der QS-Filter-Eingangsdatensatz

Der QS-Filter-Eingangsdatensatz ist je nach Richtlinien- und Abrechnungskontext zu wählen:

#### **DATENSATZ\_301 enthält den QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V**

Die Dokumentationspflicht einer Einrichtung wird ausgelöst, wenn mindestens ein Fall den definierten Auslösebedingungen entspricht. Die für den QS-Filter maßgeblichen administrativen und medizinischen Datenfelder zur Identifikation dokumentationspflichtiger Fälle im Rahmen der Qesü-RL für durch ein Krankenhaus erbrachte Fälle finden sich im Aufnahmedatensatz bzw. in der Entlassungsanzeige des § 301-Datenformats<sup>25</sup>. Daher ist der QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V (`DATENSATZ_301`) so beschaffen, dass fast alle seine Datenfelder aus

<sup>25</sup> Datenübermittlung nach § 301 Abs. 3 SGB V, ergänzt um die Festlegungen der Vereinbarung nach § 120 Abs. 3 SGB V

der Struktur des § 301-Datensatzes abgeleitet werden können. Einzige Ausnahmen sind die Felder PATALTER und FALLART (lassen sich aus Inhalten des § 301-Datensatzes berechnen).

Der QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V besteht aus folgenden Teilen:

- Behandlungsfall (Teildatensatz FALL)
- Prozedurangaben des Behandlungsfalls (Teildatensatz PROZ, wiederholbar)
- Diagnoseangaben des Behandlungsfalls (Teildatensatz DIAG, wiederholbar)
- Entgeltangaben des Behandlungsfalls (Teildatensatz ENTGELT, wiederholbar)

Die hier definierten Teildatensätze finden sich im § 301-Datensatz bzw. dem DRG-Datenformat nach § 21 KHEntgG<sup>26</sup> wieder. Der QS-Filter-Eingangsdatensatz basiert zum größten Teil auf einer Teilmenge der Inhalte dieser Datensätze.

Tabelle 14: Felder des QS-Filter-Eingangsdatensatzes nach § 301

Feld	Beschreibung	M/K	Zeichenslänge	BasisTyp
<b>Behandlungsfall (FALL)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
FALLART	Fallart 1: ENTGELTARTAMB <> LEER UND ENTGELTARTSTAT = LEER 2: ENTGELTARTSTAT <> LEER UND ENTGELTARTAMB = LEER	M	1	NUMSCHLUESSEL (Fallart)
AUFNGRUND	Aufnahmegrund gem. § 301 SGB V	K	2	SCHLUESSEL (AufnGrund)
AUFNDATUM	Datum der Aufnahme Krankenhaus	K	10	DATUM
ENTLDATUM	Datum der Entlassung Krankenhaus	K	10	DATUM
PATALTER	Alter in Jahren am Auf- nahmetag	M	3	GANZEZAHL
ENTLGRUND	Entlassungsgrund § 301	K	2	SCHLUESSEL (EntlGrund)

<sup>26</sup> Vereinbarung über die Übermittlung von DRG-Daten nach § 21 KHEntgG. Stand: 30. November 2012. Siegburg: Institut für das Entgeltsystem im Krankenhaus (InEK gGmbH). Online verfügbar unter [http://www.gdrg.de/cms/Datenlieferung\\_gem\\_21\\_KHEntgG/Dokumente\\_zur\\_Datenlieferung/Datensatzbeschreibung](http://www.gdrg.de/cms/Datenlieferung_gem_21_KHEntgG/Dokumente_zur_Datenlieferung/Datensatzbeschreibung).

Feld	Beschreibung	M/K	Zeichenlänge	BasisTyp
KASSEIKNR	Institutionskennzeichen der Krankenkasse der Versichertenkarte	K	9	GANZEZAHL
VERSICHERTENIDNEU	eGK-Versichertennummer	K	10	TEXT
PERSONENKREIS	besonderer Personenkreis	K	1	SCHLUESSEL (Personenkreis)
<b>Prozedurangaben (wiederholbar) (PROZ)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
OPDATUM	Datum der Operation	M	10	DATUM
OPS	Prozedurschlüssel (OPS-Kode)	M	13	SCHLUESSEL (OPS)
<b>Diagnoseangaben (wiederholbar) (DIAG)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
ICD	Diagnoseschlüssel (ICD-Kode)	M	9	SCHLUESSEL (ICD)
DIAGART	Diagnoseart	M	2	SCHLUESSEL (DiagArt)
SEKDIAG	Sekundär-Diagnoseschlüssel (Sekundär-Kode)	K	9	SCHLUESSEL (ICD)
<b>Entgeltangaben (wiederholbar) (ENTGELT)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
ENTGELTARTSTAT	Entgeltschlüssel stationär gem. § 301 SGB V (1.–3. Stelle)	K	3	NUMSCHLUESSEL (EntgeltSchluessel-Stat)
ENTGELTARTAMB	Entgeltschlüssel ambulant gem. § 301 SGB V	K	8	NUMSCHLUESSEL (EntgeltSchluessel-Amb)
EBMDATUM	Leistungstag (Entgelt)	K	10	DATUM

In den Datenfeldern ENTGELTARTSTAT und ENTGELTARTAMB sind die Codes zu dokumentieren, welche im Schlüssel Entgeltarten des Datenübermittlungsverfahrens nach § 301 Abs. 3 SGB V definiert sind<sup>27</sup>.

<sup>27</sup> Aktuelle Version abrufbar im Download-Bereich der DKG-Homepage (<http://www.dkgev.de>)

Für den externen Schlüssel `EntgeltSchlüsselStat` sind dreistellige Schlüsselwerte hinterlegt. Diese dienen im Rahmen der Spezifikation der Identifikation belegärztlicher Fälle.

### Primär und Sekundär Diagnosen

Seit Einführung des Verfahrens QS-WI in der Spezifikation 2017 ist das Feld `SEKDIAG` (bzw. die Syntaxvariablen `SEKDIAG` und `HSEKDIAG`) im Eingangsdatensatz enthalten. Für die jeweiligen Module sind Einschlussdiagnosen nach Primär- und Sekundär Diagnosen im Sinne der Kreuz-Stern-Notation zu differenzieren. Die Zusammengehörigkeit bestimmter Sekundär Diagnosen zu bestimmten Primär Diagnosen ist hierbei zu berücksichtigen. Die in den Auslösekriterien referenzierten Felder `DIAG` bzw. `HDIAG` beinhalten weiterhin sowohl Primär- als auch Sekundär Diagnosen.

### DATENSATZ\_295 enthält den QS-Filter-Eingangsdatensatz nach § 295 SGB V

Der QS-Filter-Eingangsdatensatz nach § 295 SGB V bezieht sich auf Fälle, die kollektivvertraglich durch einen niedergelassenen Leistungserbringer (Arztpraxis, MVZ, auch Eingriffe durch Belegärzte oder durch Krankenhausärzte im Rahmen einer KV-Ermächtigung)<sup>28</sup> erbracht werden und ist so beschaffen, dass fast alle seine Datenfelder aus der Struktur der KVDT Datensatzbeschreibung für den Einsatz von IT-Systemen in der Arztpraxis der Kassenärztlichen Bundesvereinigung gemäß § 295 SGB V abgeleitet werden können<sup>29</sup>. Einzige Ausnahme sind die Felder `PATALTER` (lässt sich aus Inhalten des KVDT-Datensatzes berechnen) und `BEHANDLFALLID` (kann zur Verknüpfung der Teildatensätze vom System vergeben werden oder auf andere Felder des KVDT-Datensatzes Bezug nehmen, sofern diese eine eindeutige Zuordnung erlauben).

Der QS-Filter-Eingangsdatensatz nach § 295 SGB V besteht aus vier Teilen:

- Behandlungsfall (Teildatensatz `FALL`)
- Prozedurangaben des Behandlungsfalls (Teildatensatz `PROZ`, wiederholbar)
- Diagnoseangaben des Behandlungsfalls (Teildatensatz `DIAG`, wiederholbar)
- Angaben zu Gebührenordnungspositionen des Behandlungsfalls (Teildatensatz `EBM`, wiederholbar)

Diese vier Teile sind in der Datenbank für den QS-Filter über separate Teildatensätze abgebildet (Tabelle `Tds`). Die einem Teildatensatz zugehörigen Datenfelder werden über die Tabelle `TdsField` definiert. Diese entsprechen den Feldern der KVDT Datensatzbeschreibung für den Einsatz von IT-Systemen in der Arztpraxis der Kassenärztlichen Bundesvereinigung gemäß § 295 SGB V oder können aus diesen abgeleitet werden. Das Attribut `TdsField.hinweis` enthält daher entsprechende Referenzen auf die Datensatzbeschreibung KVTD am Beispiel der Version 5.15 vom 07.02.2017. Die Referenz dient lediglich der Nachvollziehbarkeit der in Tabelle `TdsField` abgebildeten Datenfelder in Bezug auf die technische Anlage.

<sup>28</sup> Ambulante Leistungen nach § 295 SGB V, § 116 SGB V, stationäre Fälle nach § 121 Abs. 3 SGB.

<sup>29</sup> Zeichenart und Felddlänge der im Folgenden beschriebenen Datenfelder können vom KVDT-Datensatz abweichen, da hier die in der Spezifikation definierten Datentypen verwendet werden. Das Format `DATUM` wird beispielsweise über 10 Zeichen (`TT.MM.JJJJ`) abgebildet statt über 8 Zeichen (`JJJMMTT`).

Tabelle 15: Felder des QS-Filter-Eingangsdatensatzes nach § 295

Feld	Beschreibung Referenz Datensatzbeschreibung KVDT	M/K	Zeichen länge	BasisTyp
<b>Behandlungsfall (FALL)</b>				
SATZART	Satzart Kennung: 8000	M	4	NUMSCHLUESSEL (Satzart)
BEHANDLFALLID	Behandlungsfall-ID	M	15	TEXT
PATNUMMER	Patientennummer Kennung: 3000	K	15	TEXT
SCHEINID	Schein-ID Kennung: 3003	K	60	TEXT
GEBDATUM	Geburtsdatum Kennung: 3103	M	10	DATUM
AUFNDATUM	Aufnahmedatum Kennung: 5025	K	10	DATUM
ENTLDATUM	Entlassungsdatum Kennung: 5026	K	10	DATUM
PATALTER	Alter in Jahren am Behand- lungstag berechnetes Feld: alter(GEBDA- TUM;EBMDATUM)	M	3	GANZEZAHL
BSNRAMBULANT	Betriebs- (BSNR) oder Nebenbetriebsstättennum- mer (NBSNR) (ambulant) (N)BSNR des Ortes der Leistungs- erbringung; Kennung: 5098	M	9	SCHLUESSEL (BSNRRambulant)
LANR	Lebenslange Arztnummer (LANR) Lebenslange Arztnummer (LANR) des Vertragsarztes/Vertragspsy- chotherapeuten; Kennung: 5099	M	9	SCHLUESSEL (LANRRambulant)
KASSEIKNR	Kostenträgernummer Kostenträgerkennung; Kennung: 4111	M	9	GANZEZAHL
VERSI- CHERTENIDNEU	eGK-Versichertennummer Versicherten_ID; Kennung: 3119	K	10	TEXT
PERSONENKREIS	besonderer Personenkreis Besondere Personengruppe; Ken- nung: 4131	K	1	SCHLUESSEL (Personenkreis)
<b>Prozedurangaben (wiederholbar) (PROZ)</b>				
BEHANDLFALLID	Behandlungsfall-ID	M	15	TEXT

Feld	Beschreibung Referenz Datensatzbeschreibung KVDT	M/K	Zeichen länge	BasisTyp
OPDATUM	OP-Datum Kennung: 5034	K	10	DATUM
OPS	OP-Schlüssel Kennung: 5035	K	13	SCHLUESSEL (OPS)
OPSLOK	Seitenlokalisierung OPS Kennung: 5041	K	1	SCHLUESSEL (OPSseitenLoka- lisation)
<b>Diagnoseangaben (wiederholbar) (DIAG)</b>				
BEHANDLFALLID	Behandlungsfall-ID	M	15	TEXT
ICD	ICD-Code Kennung: 6001	M	9	SCHLUESSEL (ICD)
DIAGSICHERHEIT	Diagnosensicherheit Kennung: 6003	M	1	SCHLUESSEL (DiagSicherheit)
DIAGLOK	Seitenlokalisierung Kennung: 6004	K	1	SCHLUESSEL (ICDseitenLoka- lisation)
SEKDIAG	ICD-Code (Sekundärdiag- nose) Kennung: 6001 <sup>30</sup>	K	9	SCHLUESSEL (ICD)
<b>Entgeltangaben (wiederholbar) (EBM)</b>				
BEHANDLFALLID	Behandlungsfall-ID	M	15	TEXT
EBMDATUM	Leistungstag Kennung: 5000	M	10	DATUM
EBM	Gebührennummer gemäß EBM-Katalog GNR; Kennung: 5001	M	7	SCHLUESSEL (EBM)

### Gebührenordnungspositionen bzw. Gebührennummern

Die Gebührenordnungspositionen bzw. Gebührennummern sind nach dem gültigen einheitlichen Bewertungsmaßstab in der jeweils aktuellen Fassung zu kodieren. Für jede dokumentierte Gebührenordnungsposition ist ein neuer Teildatensatz EBM anzulegen.

<sup>30</sup> Wenn zu einer Diagnose (FK 6001) das Element „notationskennzeichen“ (SDICD) mit Inhalt „\*“ oder „!“ existiert (=Sekundärkode) [SEKDIAG], muss mind. ein ICD-Code FK 6001 ohne „notationskennzeichen“ (SDICD) oder wenn vorhanden, mit Inhalt „+“ (=Primärkode) [DIAG] vorhanden sein.

## Primär und Sekundär Diagnosen

Für die Auslösung der Module NWIES und NWIEA sind Einschlussdiagnosen nach Primär- und Sekundär Diagnosen im Sinne der Kreuz-Stern-Notation zu differenzieren. Die Zusammengehörigkeit bestimmter Sekundär Diagnosen zu bestimmten Primär Diagnosen ist hierbei zu berücksichtigen. Das in den Auslösekriterien referenzierten Feld DIAG kann sowohl Primär- als auch Sekundär Diagnosen beinhalten.

---

### Beispiel:

Die folgende Teilbedingung prüft, ob die Kombination einer **Primär Diagnose** [DIAG] aus Tabelle LISTE1\_ICD mit einer dazugehörigen **Sekundär Diagnose** [SEKDIAG] aus Tabelle LISTE2\_ICD vorliegt:

- (DIAG EINSIN LISTE1\_ICD UND SEKDIAG EINSIN LISTE2\_ICD)  
(eine Primär Diagnose aus der Tabelle LISTE1\_ICD und eine Sekundär Diagnose aus der Tabelle LISTE2\_ICD)

Um die Prüfung der Zusammengehörigkeit im Algorithmus abzubilden, steht die Teilbedingung in runden Klammern. Für die Abbildung der logischen UND-Verknüpfung wären die zusätzlichen Klammern um die Teilbedingungen jedoch nicht erforderlich.

Die folgende Teilbedingung prüft, ob eine **Primär- oder Sekundär Diagnose** [DIAG] aus Tabelle LISTE3\_ICD vorliegt:

- DIAG EINSIN LISTE3\_ICD  
(eine Diagnose aus der Tabelle LISTE3\_ICD)

---

## 1.4 Der Algorithmus zur Ermittlung der Dokumentationspflicht

Dieser Abschnitt beschreibt den Algorithmus zur Ermittlung der Dokumentationsverpflichtung von Modulen und die erforderlichen Fehlerprüfungen.

### 1.4.1 Einleitung und Überblick

Für jedes Modul ist ein formaler, logischer Ausdruck definiert, welcher sich aus einer verfahrensbezogenen Teilbedingung (Abschnitt B 1.4.2) und einer administrativen Teilbedingung (Abschnitt B 1.4.3) zusammensetzt:

Auslösebedingung für ein Modul = ModulAusloeser.bedingung UND  
AdminKriterium.bedingung

Wenn eine der beiden Teilbedingungen nicht erfüllt ist, so löst der QS-Filter für das jeweilige Modul keine Dokumentationspflicht aus.

Für jeden Fall evaluiert der QS-Filter-Algorithmus sämtliche der in der Spezifikationsdatenbank hinterlegten Auslösebedingungen. Wird die Auslösebedingung erfüllt, so ist der Fall für das verknüpfte Modul dokumentationspflichtig.

**Achtung**

Generell gilt die Einschränkung, dass ein gleiches Modul pro Einrichtung nur einmal ausgelöst wird. Es können aber mehrere verschiedenartige Module ausgelöst werden. Ein Krankenhaus kann beispielsweise als stationäre Einrichtung für das Modul NWIES und als ambulant operierende Einrichtung für das Modul NWIEA dokumentationspflichtig werden.

---

**Dokumentationspflicht einer Einrichtung**

Eine Einrichtung ist dokumentationspflichtig, wenn für sie mindestens ein QS-Modul dokumentationspflichtig ist. Ein QS-Modul wird als dokumentationspflichtig identifiziert, wenn mindestens ein Fall den definierten Auslösekriterien entspricht.

**Dokumentationspflicht eines Moduls**

Ein Modul (Datensatz) ist dann dokumentationspflichtig, wenn folgende Bedingungen zutreffen:

- Bei der Fehlerprüfung tritt kein Fehler auf (Abschnitt B 1.4.5).
- Die jeweilige modulspezifische Auslösebedingung ist erfüllt (Abschnitt B 1.4.2).
- Die administrativen Bedingungen sind erfüllt (Abschnitt B 1.4.3).

**1.4.2 Verfahrensbezogene Einschlusskriterien****Überblick:**

Für jede QS-Datensatz-Definition (= Modul) sind in der Tabelle ModulAuslöser eine oder mehrere Auslösebedingungen (Attribut bedingung) hinterlegt.

**Definition:**

Jede Auslösebedingung der Tabelle ModulAuslöser definiert einen verfahrensspezifischen QS-Filter-Modulauslöser.

Ein QS-Filter-Modulauslöser bildet somit ein Komplex von Leistungen ab, welche über den QS-Filter-Algorithmus zu einer Dokumentationsverpflichtung führen.

---

**Achtung****Einschlusskriterien mit KASSEIKNR, VERSICHERTENIDNEU und PERSONENKREIS**

Verfahrensbezogene Einschlusskriterien können die Felder KASSEIKNR, VERSICHERTENIDNEU und PERSONENKREIS enthalten, mithilfe dessen geprüft wird, ob es sich um einen GKV-Versicherten handelt. Liegt die elektronische Gesundheitskarte (eGK) bei der Prüfung der definierten Kriterien noch nicht vor, erlischt **nicht** die Dokumentationspflicht. Die Auslösebedingung ist bei Vorliegen der Angaben KASSEIKNR/VERSICHERTENIDNEU/PERSONENKREIS/eGK erneut zu prüfen.

---

Ein QS-Filter-Modulauslöser identifiziert in Kombination mit dem administrativen Kriterium die Dokumentationspflicht für immer genau einen QS-Datensatz.

- Die QS-Filter-Modulauslöser NWIEA\_LKG und NWIEA\_KV lösen jeweils den QS-Datensatz

NWIEA aus.

- Die QS-Filter-Modulauslöser `NWIES_LKG` und `NWIES_KV` lösen jeweils den QS-Datensatz `NWIES` aus.

### Definition in der Datenbank

Tabelle 16 gibt einen Überblick über die Tabelle `ModulAusloeser` der Spezifikationsdatenbank.

Tabelle 16: Struktur der Tabelle `ModulAusloeser`

Feldname	Datentyp	Bemerkung
<code>idModulAusloeser</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name des QS-Filter-Modulauslösers
<code>bezeichnung</code>	TEXT	Bezeichnungstext der Auslösebedingung
<code>bedingung</code>	MEMO	Auslösebedingung für den QS-Filter-Modulauslöser
<code>textDefinition</code>	MEMO	Medizinisch-inhaltliche Definition bzw. Erläuterung der Auslösebedingung
<code>fkModul</code>	INTEGER	Bezug zum Modul
<code>verpflichtend</code>	BOOLEAN	Wenn WAHR, ist das betreffende Modul bundesweit verpflichtend
<code>fkAdminKriterium</code>	INTEGER	Bezug zu den administrativen Abgrenzungskriterien
<code>fkDs</code>	INTEGER	Bezug zum <code>DATENSATZ_301</code> oder zum <code>DATENSATZ_295</code>
<code>internBild</code>	TEXT	Bild zum Modul
<code>ambulant</code>	BOOLEAN	Ist die Fallart ambulant (bezogen auf Fälle einer Einrichtung)?
<code>stationaer</code>	BOOLEAN	Ist die Fallart stationär (bezogen auf Fälle einer Einrichtung)?
<code>krankenhausabrechnung</code>	BOOLEAN	Wird der Fall stationär abgerechnet (bezogen auf Fälle einer Einrichtung)?
<code>selektiv</code>	BOOLEAN	Wird der Fall ambulant selektivvertraglich abgerechnet (bezogen auf Fälle einer Einrichtung)?
<code>kollektiv</code>	BOOLEAN	Wird der Fall ambulant kollektivvertraglich abgerechnet (bezogen auf Fälle einer Einrichtung)?

## Identifizierung der Fallart und der Art der Leistungserbringung

Zur Abbildung sektorenübergreifender Module wurden die booleschen Attribute zur Fallart (`ambulant`, `stationaer`) und zur Art der Leistungserbringung (`krankenhausabrechnung`, `kollektiv`, `selektiv`) in die Tabelle `ModulAusloeser` integriert. Diese können wie folgt identifiziert werden:

### **ambulant**

DATENSATZ\_301 : Dieses Attribut ist WAHR, wenn Feld `FALLART` = 1 .

DATENSATZ\_295 : Dieses Attribut ist WAHR, wenn Feld `SATZART` = '0101' .



#### **Hinweis zu DATENSATZ\_295**

Anstelle der `SATZART` dürfen zur Differenzierung zwischen `ambulant` und `stationaer` auch andere im jeweiligen Abrechnungssystem vorliegenden identifizierende Merkmale (z.B. die Betriebsstättennummer) von den Kassenärztlichen Vereinigungen herangezogen werden.

Die Prüfung muss hierbei jederzeit zum selben Ergebnis führen wie die Prüfung der `SATZART`.

---

### **stationaer**

DATENSATZ\_301 : Dieses Attribut ist WAHR, wenn Feld `FALLART` = 2 .

DATENSATZ\_295 : Dieses Attribut ist WAHR, wenn Feld `SATZART` = '0103' .



#### **Hinweis zu DATENSATZ\_295**

Anstelle der `SATZART` dürfen zur Differenzierung zwischen `ambulant` und `stationaer` auch andere im jeweiligen Abrechnungssystem vorliegenden identifizierende Merkmale (z.B. die Betriebsstättennummer) von den Kassenärztlichen Vereinigungen herangezogen werden.

Die Prüfung muss hierbei jederzeit zum selben Ergebnis führen wie die Prüfung der `SATZART`.

---

### **krankenhausabrechnung**

Dieses Attribut ist WAHR, wenn Feld `ENTGELTARTAMB` <> LEER ODER (`ENTGELTARTSTAT` <> LEER UND `ENTGELTARTSTAT` KEINSIN `STAT_BELEG_ENTGELT`).

### **kollektiv/selektiv<sup>31</sup>**

Diese Attribute können über das Feld `EBM` identifiziert werden. Das Attribut `selektiv/kollektiv` ist WAHR, wenn `EBM` <> LEER. In diesem Fall ist das Attribut `krankenhausabrechnung` = FALSCH.

---

<sup>31</sup> Die Attribute `kollektiv` und `selektiv` sollen die Unterscheidung beider Attribute ermöglichen. Die Attribute können ggf. nicht durch jeden Anbieter differenziert werden. Eine spätere Konkretisierung wird in der Spezifikation vorgenommen, sobald eindeutige Marker existieren bzw. identifiziert werden konnten, die eine bundesweite Differenzierung zwischen selektivvertraglichen und kollektivvertraglichen Fällen ermöglichen.

**Hinweis zu DATENSATZ\_295**

Die Auslösung der einrichtungsbezogene QS-Dokumentation Einrichtungen erfolgt auf Grundlage von Abrechnungsdaten bei den Kassenärztlichen Vereinigungen. Das Attribut `kollektiv` ist in diesem Fall immer `TRUE`.

---

**1.4.3 Administrative Einschlusskriterien****Überblick**

Die administrativen Regelungen beschreiben diejenigen Auslösebedingungen, welche über die modulspezifischen ICD/OPS/EBM-Kriterien (Abschnitt B 1.4.2) hinausgehen. Moduldokumentationen werden für eine Einrichtung (Prüfung auf Fallebene) nur dann ausgelöst, wenn

1. der Patient stationär/ambulant aufgenommen/behandelt wird,
2. der Behandlungsfall einen bestimmten zeitlichen Rahmen aufweist,
3. der Behandlungsfall keine Verbringungsleistung für Dritte ist.

**Zu 1.**

Stationär aufgenommen: Die Bedingung wird aus der Angabe im Datenfeld `Aufnahmegrund` (Abschnitt B 1.3.1) abgeleitet:

```
AUFNGRUND NICHTIN ('03';'04') UND AUFNGRUND <> LEER
```

Diese Teilbedingung ist Teil der administrativen Kriterien (siehe Attribut `bedingung` in Tabelle 17). Der zeitliche Rahmen wird bei stationären Fällen über das Aufnahme- und das Entlassungsdatum definiert.

**Zu 2.<sup>32</sup>**

---

**Beispiel:**

Stationäre Patienten, welche im 1. oder 2. Quartal 2017 des Jahres 2017 aufgenommen und entlassen sind:

```
AUFNDATUM >= '01.01.2017' UND AUFNDATUM <= '30.06.2017' UND  
ENTL DATUM >= '01.01.2017' UND ENTL DATUM <= '30.06.2017'
```

---

**Zu 3.**

Dieses Kriterium ist nicht aus ein oder mehreren Datenfeldern des QS-Filter-Eingangsdatensatzes ableitbar. Hier ist seitens des Krankenhauses für eine organisatorische Regelung zu sorgen.

**Definition in der Datenbank**

Tabelle 17 gibt einen Überblick über wichtige Attribute der Tabelle `AdminKriterium`, welche die administrativen Abgrenzungskriterien definiert.

---

<sup>32</sup> Verbindlich sind die in der Spezifikationsdatenbank hinterlegten Kriterien.

Tabelle 17: Struktur der Tabelle AdminKriterium

Feldname	Datentyp	Bemerkung
idAdminKriterium	INTEGER	Primärschlüssel
name	TEXT	Technischer Name des administrativen Kriteriums
bedingung	MEMO	Administrative Auslösebedingung
info	TEXT	Bezeichnungstext des administrativen Kriteriums
sollBedingung	TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
sollBedingungUeberlieger	TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.

#### 1.4.4 Struktur und Syntax der Auslösebedingungen

##### Die Variablen der Auslösebedingungen

Die in den Auslösebedingungen erlaubten Variablen sind in der Tabelle `SyntaxVariable` definiert.

Die Variablennamen (Attribut `SyntaxVariable.name`) bestehen aus maximal 32 Zeichen. Sie dürfen nur die Buchstaben A bis Z (Großbuchstaben) und die Ziffern 0 bis 9 enthalten. Ein Feldname muss immer mit einem Buchstaben beginnen. Umlaute und Sonderzeichen sind in Feldnamen nicht erlaubt. Ein Feldname darf auch nicht ein reserviertes Wort sein (z.B. Namen von Operatoren wie `EINSIN`, siehe Tabelle 19).

##### Typen

Jede Variable hat einen Basistyp. Tabelle 18 gibt einen Überblick über die möglichen Basistypen. Jeder der in der Tabelle `SyntaxVariable` definierten Variablen ist über den Wert des Attributs `SyntaxVariable.fkTdsFeld` ein Feld des QS-Filter-Eingangsdatensatzes zugeordnet. Jedes dieser Felder besitzt einen Basistyp.

Tabelle 18: Basistypen der Variablen

Basistyp	Bezeichnung	Beispiele (Literals)
BOOL	Boolesche Variable	WAHR, FALSCH
TEXT	Zeichenkette (String)	„Spezifikation“
GANZEZAHL	... -2, -1, 0, 1, 2, 3, ...	1
ZAHL	Zahl (mit oder ohne Nachkommastellen)	Erfassung: 25, 4 oder -100, 8

Basistyp	Bezeichnung	Beispiele (Literele)
		Export: 25 . 4 oder 100 . 8
DATUM	Zehnstelliges Datum	'01.01.2013'
NUMSCHLUESSEL	Numerisch kodierter Schlüssel (wie GANZEZAHL)	0
SCHLUESSEL	Alphanumerischer Schlüssel	'19.1', '07'
UHRZEIT	Uhrzeit	'10:15'
JAHRDATUM	Jahresdatum	2017
QUARTDATUM	Quartalsdatum (ohne Angabe von Tagen oder Monaten)	'1/2017'

Die meisten Schlüsselwerte werden als GANZEZAHL kodiert, d. h., dass die Codes nicht in Hochkommata gesetzt werden dürfen. Ausnahmen sind z.B. die OPS-Schlüssel (z.B. '5.282.0') oder die ICD-10-Schlüssel (z.B. 'C11.3').

### Listen

Eine Variable wird als Liste interpretiert, wenn der Wert des Attributs `SyntaxVariable.istListe = WAHR` ist.

### Literele

Alphanumerische Literale (z.B. SCHLUESSEL) werden von einfachen Hochkommata eingeschlossen, während Zeichenketten vom Datentyp TEXT in doppelte Anführungsstriche gesetzt werden müssen.

Dies gilt nicht für numerischen Literale (GANZEZAHL, ZAHL, NUMSCHLUESSEL) und Literale des Datentyps BOOL (Wahrheitswerte).

### Diagnose-, Prozeduren-, Abrechnungs- und Fachabteilungslisten

Fest definierte Listen von Diagnosen, Prozeduren oder Abrechnungsziffern werden in der Syntax ähnlich wie die Variable `TRACER_OPS_STAT` angesprochen.

Die erlaubten Namen von Diagnoselisten sind in der Tabelle `ICDListe` (Attribut name) zu finden. Analog sind die zulässigen Namen von Prozedurlisten in der Tabelle `OPSListe` (Attribut name) und von Abrechnungslisten in der Tabelle `EBMListe` bzw. `EntgeltListe` (Attribut name) definiert. Die Namen der Fachabteilungslisten sind in der Tabelle `FABListe` zu entnehmen.

Die jeweiligen Codes (nach ICD-10-GM, OPS bzw. EBM) sind in den durch Fremdschlüsselbeziehungen verknüpften Tabellen `ICDWert`, `OPSWert` und `EBMWert` enthalten. Alle Codes sind terminal. Die Diagnosecodes enthalten weder die Suffixe +, \* oder ! noch die Zusatzkennzeichen zur Diagnosesicherheit A, G, V und Z.

Die Codes zu Entgelten und Fachabteilungen sind in den durch Fremdschlüsselbeziehungen verknüpften Tabellen `EntgeltWert` und `FABWert` enthalten.

### Die Operatoren der Auslösebedingungen

Tabelle 19 gibt einen Überblick über die in der Syntax zulässigen Operatoren. Der aktuelle Überblick über alle zulässigen Operationen (inkl. Operanden) ist in Tabelle `SyntaxOperator` der QS-Filter-Datenbank zu finden.

Tabelle 19: Präzedenz und Assoziativität der Operatoren<sup>33</sup>

Präzedenz	Assoziativität	Operator	Erläuterung
0	links	EINSIN	Mengenoperator „EINSIN“
	links	KEINSIN	Mengenoperator „KEINSIN“
3	links	<	Vergleichsoperator „kleiner“
	links	>	Vergleichsoperator „größer“
	links	<=	Vergleichsoperator „kleinergleich“
	links	>=	Vergleichsoperator „größergleich“
4	links	=	Vergleichsoperator „gleich“
	links	<>	Vergleichsoperator „ungleich“
5	rechts	NICHT	Logischer Operator „NICHT“
6	links	UND	Logischer Operator „UND“
7	links	ODER	Logischer Operator „ODER“

#### Anmerkungen:

Operatoren mit beidseitigen Listenfeldern als Operanden:

**EINSIN:** Wenn mindestens ein Element aus der linken Liste in der rechten Liste enthalten ist, so ist der Ausdruck wahr (nichtleere Schnittmenge).

**KEINSIN:** Wenn kein Element der linken Liste in der rechten Liste enthalten ist, so ist der Ausdruck wahr (leere Schnittmenge)<sup>34</sup>.

#### Beispiel:

Folgende Regel prüft, ob ein Element der Listenvariable `PROZ` einen bestimmten Code besitzt:

```
PROZ EINSIN TRACER_OPS_AMB
```

<sup>33</sup> In dieser Übersichtstafel hat jeder einzelne Operator eine Präzedenzstufe (höchste Präzedenzstufe ist 0). Operatoren, welche die gleiche Stufe haben, werden nach den Regeln der Assoziativität aufgelöst.

<sup>34</sup> Dieser Operator ist redundant, da er auch durch Negation des `EINSIN`-Operators abgedeckt ist.

### String-Vergleich bei Diagnosen, Prozeduren und Abrechnungsziffern

Die Operation `EINSIN` basiert für Diagnosen, Prozeduren oder Abrechnungsziffern auf einem Vergleich von Zeichenketten (String-Vergleich): Für jedes Element der linken Liste wird über einen String-Vergleich geprüft, ob es in der rechten Liste enthalten ist.

Eine Modifikation des String-Vergleichs ist beim String-Vergleich von Diagnosen nötig: Hier werden die Suffixe `+`, `*` und `!` sowie die Zusatzkennzeichen `A`, `G`, `V` und `Z` ignoriert. Nicht ignoriert wird das Sonderzeichen `.` (und `-` bei OPS).



#### Achtung

In der QS-Filter-Datenbank sind nur terminale Codes enthalten. Entsprechend muss sichergestellt sein, dass auch die Codes aus dem Eingangsdatensatz terminal sind. Darauf ist besonders zu achten, wenn Daten ungeprüft aus Subsystemen übernommen wurden.

#### Beispiel:

Folgende Operationen führen zum gleichen Ergebnis

```
('Z37.9') EINSIN ('Z37.9'; 'Z37.0'; 'Z37.1'; 'Z37.2'; 'Z37.3')
('Z37.9!' ) EINSIN ('Z37.9'; 'Z37.0'; 'Z37.1'; 'Z37.2'; 'Z37.3')
```

Eine analoge Regelung gilt für die Seitenlokalisationen der OPS-Kodes.

#### 1.4.5 Fehlerprüfung

Vor Evaluation der in den vorhergehenden Abschnitten beschriebenen Auslösebedingungen ist eine Fehlerprüfung durchzuführen. Die Fehlerprüfung bezieht sich auf die Daten des QS-Filter-Eingangsdatensatzes. Das Ergebnis der Fehlerprüfung wird im Teildatensatz `FEHLER` gespeichert.

Tabelle 20: Fehlercodes des QS-Filters

Beschreibung des Fehlers	Fehlercode	Formale Fehlerbedingung	Fehlermeldung
Überprüfung des Formats	1	Abschnitt B 2.3.8	Der Wert ' <code>&lt;WERT&gt;</code> ' des Datenfeldes <code>&lt;Feld.name&gt;</code> ' <code>&lt;TdsFeld.bezeichnung&gt;</code> ' ist kein gültiger <code>&lt;BasisTyp.name&gt;</code> -Wert ( <code>&lt;BasisTyp.bezeichnung&gt;</code> <code>&lt;BasisTyp.format&gt;</code> ).
Überprüfung der Feldlänge	2	Abschnitt B 2.3.8	Der Wert ' <code>&lt;WERT&gt;</code> ' des Datenfeldes <code>&lt;Feld.name&gt;</code>

Beschreibung des Fehlers	Fehler- kode	Formale Fehlerbedingung	Fehlermeldung
			' <TdsFeld.bezeichnung> ' überschreitet die zulässige Feldlänge <Feld.laenge>.
Sind in den Datenfeldern mit internen und externen Schlüsseln (Basistyp SCHLUESSEL oder NUM-SCHLUESSEL) gültige Schlüsselkodes eingetragen? <sup>35</sup>	3	Abschnitt B 2.3.8	Ungültiger Schlüsselcode <Wert> des Schlüssels <Schluessel.name> im Datenfeld <Feld.name> ' <TdsFeld.bezeichnung> ' !
Überprüfung numerischer Wertebereiche	4	Abschnitt B 2.3.8	Der Wert ' <WERT> ' des Datenfeldes <Feld.name> ' <TdsFeld.bezeichnung> ' ist kleiner als ' <Feld.min> ' oder Der Wert ' <WERT> ' des Datenfeldes <Feld.name> ' <TdsFeld.bezeichnung> ' ist größer als ' <Feld.max> '
Ist ein Muss-Feld ausgefüllt?	5	Abschnitt B 2.3.8 : Attribut <TdsFeld.fkMussKann>	Das Datenfeld <Feld.name> ' <TdsFeld.bezeichnung> ' muss einen gültigen Wert enthalten.

<sup>35</sup> Die Korrektheit (z.B. ausschließlich gültige terminale Codes oder Verwendung gültiger Katalogversionen) der extern definierten Codes (ICD und OPS) muss vor der Prüfung des Datensatzes sichergestellt sein.

Ist der Fall (der Einrichtung) dem Erfassungsjahr der Spezifikation zugeordnet? (Spezifikation 2017)	6	<pre>(MODUL = 'NWIES' UND FALLART = 2 UND (AUFNDATUM &lt;'01.01.2017' ODER AUFNDATUM&gt; '30.06.2017') ) O- DER (MODUL = 'NWIEA' UND FALL- ART = 1 UND quar- talswert- Liste(OPDATUM) KEINSIN ('1/2017';2/2017') ) ODER ((MODUL = 'NWIES' UND SATZ- ART = '0103' ) O- DER (MODUL = 'NWIEA' UND SATZ- ART = '0101' )) UND quartalswert- Liste(EBMDATUM) KEINSIN ('1/2017';2/2017') )</pre>	Die Einrichtung ist im Jahr 2017 nicht dokumentationspflichtig: Aufnahmedatum/OP-Datum/EBM-Datum= <Wert>
--	---	---	--

In <Wert> ist der Wert des Datenfeldes der Falldaten einzusetzen, auf den sich die Fehlermeldung bezieht. Ansonsten sind in die <...> -Ausdrücke die entsprechenden Einträge aus der Datenbank einzusetzen. Es wird beim Auftreten eines Fehlers nicht weiter geprüft, ob für den Behandlungsfall einer Einrichtung Moduldokumentationen ausgelöst werden.

## 2 QS-Dokumentation

Die vorliegenden Spezifikationskomponenten für die QS-Dokumentation definieren die Vorgaben zur Datenerfassung, Plausibilitätsprüfung und Datenübermittlung für die externe vergleichende Qualitätssicherung im Rahmen der Qesü-RL. Diese sollen die Bereitstellung valider und vergleichbarer Daten gewährleisten. Neben der Datenbank zur QS-Dokumentation zählen zu den Spezifikationskomponenten der QS-Dokumentation die Ausfüllhinweise und die Dokumentationsbögen. Die Delta-Tabellen in der Datenbank zur QS-Dokumentation stellen geänderte, gelöschte und neue Datenbankinhalte im Vergleich zur letzten gültigen Version des Vorjahres sowie, sofern vorhanden, zur letzten Version des aktuellen Erfassungsjahres dar.

### 2.1 Anmerkungen zur Struktur der Spezifikation zur QS-Dokumentation

Die Spezifikation zur QS-Dokumentation ist in einer relationalen Datenbank abgelegt. Zurzeit wird sie ausschließlich als Access-Datenbank (MS Access 2007-2013) zur Verfügung gestellt. Der Name der Spezifikation richtet sich nach folgendem Schema:

`<Erfassungsjahr>_Qesue_QSDOK_EDOK_V<Versionsnummer>.mdb`

`<Erfassungsjahr>` bezeichnet das Jahr, in dem die QS-Dokumentation stattfindet. `<Versionsnummer>` bezeichnet die 2-stellige Versionsnummer (z.B. 01).

---

#### Beispiel:

Im Erfassungsjahr 2017 ist die Spezifikation `2017_Qesue_QSDOK_EDOK_V01.mdb`<sup>36</sup> gültig.

---

Weiterführende Erläuterungen zum Benennungsschema für Spezifikationskomponenten sind der Einleitung, Abschnitt 1.1.2, zu entnehmen. Die Tabellenstruktur der Spezifikationsdatenbank wird in Abschnitt B Komponenten beschrieben.

Folgende Attribute treten in vielen Tabellen auf:

- `name` ist in der Regel als technischer Name zu verstehen. Zum Beispiel wird `Field.name` als Variablenname in den Plausibilitätsregeln verwendet.
- `bezeichnung` ist eine kurze Beschreibung. Zum Beispiel ist `BogenField.bezeichnung` der Text, welcher ein Feld auf einem Eingabeformular beschreibt.
- `bedingung` enthält einen logischen Ausdruck. Prominentester Vertreter dieses Attributtyps ist das Attribut `bedingung` in der Tabelle `Regeln`.

### 2.2 Datenfeldbeschreibung

Für jedes Modul existiert eine eigene Datenfeldbeschreibung. Sie spezifiziert alle auszufüllenden Datenfelder (Bogenfelder, auch Items genannt) und besteht aus mehreren Tabellen (Abbildung 3), die in den nachfolgenden Abschnitten erläutert werden.

---

<sup>36</sup> Die Versionsnummer der gültigen Spezifikation (z.B. V01, V02 usw.) ist dem zuletzt veröffentlichten Update zu entnehmen.

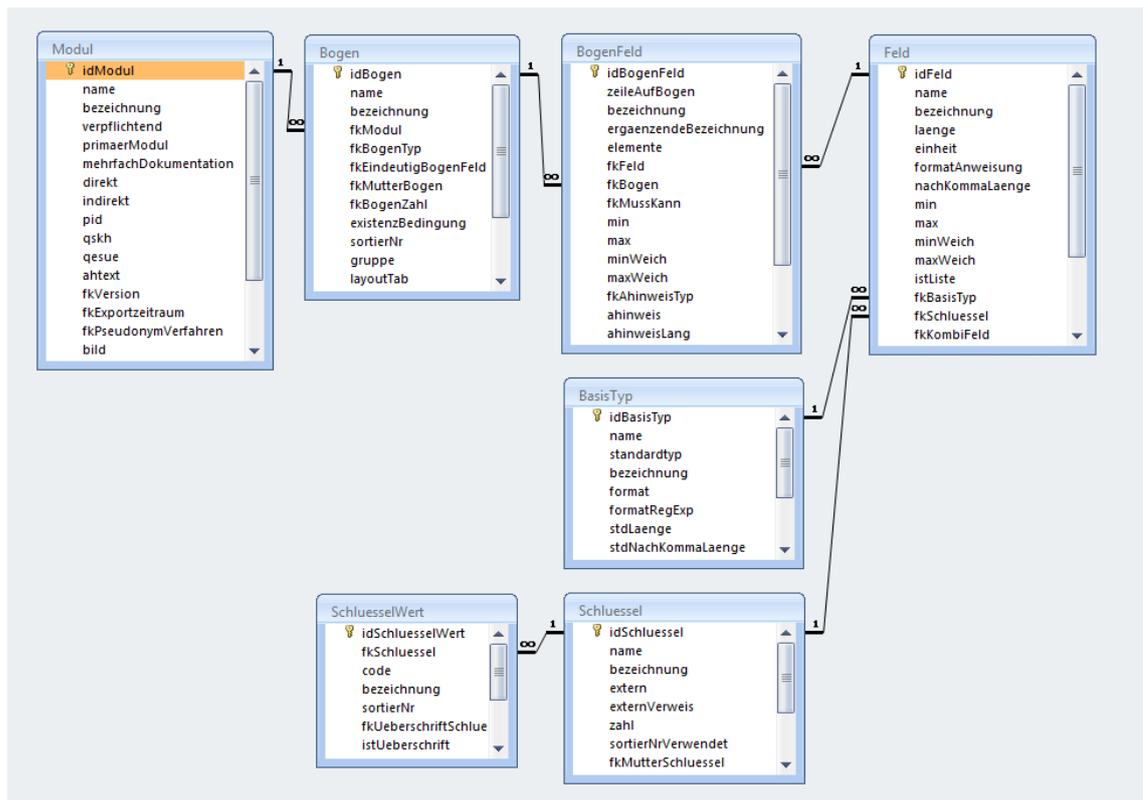


Abbildung 3: Tabellen und Relationen der Datenfeldbeschreibung

Die Abfragen Datenfeldbeschreibung und DatenfeldbeschreibungFürEinModul der Access-Datenbank ermöglichen den Überblick über diese Struktur.

Die Beschreibung der Datenfelder hat folgende Ziele:

- Bereitstellung der Informationen, welche für die Programmierung des Eingabeformulars und für die Sicherung der eingegebenen Daten nötig sind
- Vermeidung von Redundanzen
- Typisierung der Felder nach fachlichen und datentechnischen Kriterien

Das für den Anwender wichtigste Merkmal ist die Bezeichnung des Datenfelds (Attribut BogenFeld.bezeichnung).

Die Datenfeldbeschreibung ist teilweise auf dem jeweiligen Dokumentationsbogen eines Moduls („Bogensicht“) abgebildet. Zu beachten ist dabei, dass die „Bogensicht“ lediglich die Sicht der medizinischen Fachgruppen, die die Module entwickeln, darstellt. Bei verteilten Softwarelösungen<sup>37</sup> für einen Leistungserbringer hingegen ist die Bogensicht dann nicht mehr adäquat, wenn die Bestandteile eines Bogens auf verschiedene Teilsysteme verteilt sind. Die Daten eines Bogens werden in diesen Fällen für den Export aus den einzelnen Teilsystemen zusammengestellt.

<sup>37</sup> Die Kassenärztlichen Vereinigungen als DAS stellen den Vertragsärzten in ihrem Zuständigkeitsbereich Anwendungen zur Erfassung und Übertragung von Daten der einrichtungsbezogenen QS-Dokumentation, z.B. webbasiert, zur Verfügung (ggf. mit Unterstützung durch die Kassenärztliche Bundesvereinigung). Im Krankenhaussektor wird von Softwareherstellern eine QS-Dokumentationssoftware umgesetzt.

**Hinweis**

Die Papierbögen werden lediglich zu Illustrationszwecken zur Verfügung gestellt. Sie sind zur Dokumentation nicht zugelassen.

Im Kontext einer integrierten, prozessorientierten Dokumentationssoftware müssen die Teildatensätze nicht direkt in Eingabeformulare umgesetzt werden. Es ist sinnvoller, die Teile eines Dokumentationsbogens zu dem Zeitpunkt und in dem Dokumentationskontext zu erfragen, der sich in den Prozessablauf<sup>38</sup> eines Leistungserbringers einordnet.

### 2.2.1 Dokumentationsmodule (Datensätze)

Ein Modul der Spezifikation enthält die Datensatzdefinition von einem oder mehreren Verfahren (Qesü-RL). Abhängig von (inhaltlich oder organisatorisch) abzugrenzenden Bereichen kann ein Verfahren mehrere Module umfassen. Für eine Einrichtung kann eine oder mehrere Moduldokumentationen angelegt werden, die nach Dokumentationsabschluss an die Datenannahmestelle übermittelt werden. Fehlerfreie Moduldokumentationen (verkürzt „Module“), die die Basis der Datenauswertungen bilden, werden dem Leistungserbringer von der Datenannahmestelle bestätigt. Aus technischer Sicht ist ein Modul durch einen eindeutigen Namen gekennzeichnet. Es umfasst mindestens einen Teildatensatz. In der Tabelle `Modul` der QS-Spezifikation finden sich die zentralen Definitionen eines Moduls.

Tabelle 21: Struktur der Tabelle `Modul`

Feldname	Datentyp	Bemerkung
<code>idModul</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT ( 32 )	Eindeutiger technischer Name
<code>bezeichnung</code>	TEXT ( 255 )	Erläuternde Bezeichnung
<code>verpflichtend</code>	BOOLEAN	Besteht für das Modul eine QS-Dokumentationsverpflichtung?
<code>primaerModul</code>	BOOLEAN	Ist das Modul ein Primärmodul?
<code>mehrfachDokumentation</code>	BOOLEAN	Ist ein mehrfaches Anlegen eines gleichartigen Datensatzes pro Einrichtung zulässig (WAHR / FALSCH)?  Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation. Eine Mehrfachdokumentation ist jederzeit unzulässig.
<code>direkt</code>	BOOLEAN	Handelt es sich um ein direktes Datenexportverfahren?

<sup>38</sup> Zum Beispiel in den Prozessablauf eines Krankenhauses.

Feldname	Datentyp	Bemerkung
		Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
indirekt	BOOLEAN	Handelt es sich um ein indirektes Datenexportverfahren?
pid	BOOLEAN	Handelt es sich um ein Modul zur Follow-up-Erhebung? <sup>39</sup> Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
qskh	BOOLEAN	Handelt es sich um ein Modul der QSKH-Richtlinie? Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
gesue	BOOLEAN	Handelt es sich um ein Modul der Qesü-Richtlinie?
ahtext	MEMO	Einleitender Text für den Ausfüllhinweis eines Moduls
fkVersion	INTEGER	Gültige Version des jeweiligen Moduls
fkExportzeitraum	INTEGER	Exportzeitraum ( <code>Exportzeitraum.exportBis</code> ) eines Moduls
fkPseudonymVerfahren	INTEGER	Pseudonymisierung von Verfahren
Bild	TEXT ( 20 )	Modulspezifisches Bild

In der Tabelle `Modul` wird der Exportzeitraum für ein Modul angegeben (Attribut `Modul.fkExportzeitraum`). Die hinterlegten Zeitpunkte sind in der Tabelle `Exportzeitraum` (Attribut `exportBis`) definiert.

### Auslösung der Moduldokumentation

Der auslösende Sachverhalt für die Dokumentationspflicht ist in der Spezifikation für den QS-Filter definiert. Dieser greift zu diesem Zweck unter anderem auf ICD- und OPS-Kodes (z.B. Haupt- und Nebendiagnosen und Prozeduren) und Gebührenordnungspositionen (EBM-Kodes) zurück, die z.B. im Krankenhausinformationssystem (KIS) oder in den Abrechnungsdaten der Kassenärztlichen Vereinigungen verfügbar sind. Die Dokumentationspflicht der Module des Verfahrens QS-WI wird unter anderem über Einschlussdiagnosen ermittelt, die nach Primär- und Sekundär Diagnosen im Sinne der Kreuz-Stern-Notation zu differenzieren sind. Die

<sup>39</sup> Gemeint ist eine Follow-up-Erhebung mittels PID-Verfahren.

Zusammengehörigkeit bestimmter Sekundär Diagnosen zu bestimmten Primär Diagnosen ist bei der Auslösung zu berücksichtigen.

### Primärmodule

Für Primärmodule sind in der Spezifikation für QS-Filter-Software Auslösebedingungen definiert. In der vorliegenden Spezifikation der einrichtungsbezogenen QS-Dokumentation sind alle Module als Primärmodule gekennzeichnet.

### 2.2.2 Teildatensätze

Die Begriffe „Teildatensatz“ und „Bogen“ werden synonym gebraucht. In den der Illustration dienenden Dokumentationsbögen werden alle Teildatensätze aufgeführt.

#### Benennung von Teildatensätzen

Ein Teildatensatz wird durch die folgende Kombination von Modulnamen und Bogennamen identifiziert und angesprochen:

<Modul.name> : <Bogen.name>

---

#### Beispiele:

NWIEA : Bist der Basisbogen des Moduls Vermeidung nosokomialer Infektionen:  
Postoperative Wundinfektionen (einrichtungsbezogen ambulant)

---

#### Bogentyp

Der Kerndatensatz besteht aus mindestens einem Basisteildatensatz und kann durch einen oder mehrere Teildatensätze ergänzt werden. Das Attribut `Bogen.fkBogenTyp` definiert für jeden Teildatensatz seine Rolle im und seine Zugehörigkeit zum Kerndatensatz. In Tabelle 22 sind die Bezeichnungen der einzelnen Bogentypen definiert.

Tabelle 22: Inhalte der Tabelle `BogenTyp`

idBogenTyp	Bezeichnung
B	Basisteildatensatz (Teil des Kerndatensatzes)
K	Teildatensatz ist Teil des Kerndatensatzes und kein Basisteildatensatzes
O	Teildatensatz ist Teil des optionalen Datensatzes

Die Module NWIES und NWIEA verfügen jeweils nur über einen Basisteildatensatz.

Weiterführende Erläuterungen zu Teildatensätzen sind in der Technischen Dokumentation zur Basisspezifikation wird Leistungserbringer zu entnehmen.

### 2.2.3 Datenfelder (Bogenfelder)

Jedes auf einem Teildatensatz vorhandene und auszufüllende Feld wird als Datenfeld (Item, Bogenfeld) bezeichnet. Datenfelder sind charakterisiert durch ihren Namen (Bezeichnung) und die Spezifikation des einzutragenden Inhalts.

Die Bezeichnung<sup>40</sup> wird so gewählt, dass sie einem medizinischen Experten unmittelbar verständlich ist. Die Spezifikation des Inhalts umfasst hingegen sowohl eine fachliche (medizinische) als auch datentechnische Typisierung. Dagegen repräsentieren die in der Tabelle `Feld` aufgelisteten Felder inhaltlich gleiche Dokumentationsfelder mehrerer Module (Abschnitt B 2.2.1), der datentechnische Typ (`BasisTyp`) charakterisiert das Format des Feldes (Abschnitt B 2.2.1).

Jedes Datenfeld hat zwingend einen Bezug zu einem Teildatensatz und zu einem technischen Feld. Weitere Eigenschaften sind die Bogenfeldbezeichnung und die fortlaufende Nummer im Teildatensatz. Die Datenfelder sind in der Tabelle `BogenFeld` gespeichert.

Identifizierendes Merkmal eines Datenfelds ist eine Kombination aus `fkBogen` und `fkFeld`. Das bedeutet, dass das Datenbankschema gewährleistet, dass der technische Feldname (`Feld.name`) in einem Teildatensatz maximal einmal vorkommt. Per definitionem muss ein Datenfeldname sogar innerhalb eines Moduls eindeutig sein, d.h. dass eine Abfrage mit dem Primärschlüsselpaar (`modulNr, fieldNr`) genau einen Primärschlüssel `idBogenFeld` liefert.

Tabelle 23: Struktur der Tabelle `BogenFeld`

Feldname	Datentyp	Bemerkung
<code>idBogenFeld</code>	INTEGER	Primärschlüssel
<code>zeileAufBogen</code>	DOUBLE	Zeile auf dem Dokumentationsbogen.
<code>bezeichnung</code>	TEXT	Beschreibender Text zum Feld auf dem Dokumentationsbogen. Wenn der Feldinhalt leer ist, so wird der Inhalt des gleichnamigen Felds in der Tabelle <code>Feld</code> genommen.
<code>ergaenzendeBezeichnung</code>	TEXT	Optionale ergänzende Bezeichnung zu einem Bogenfeld.
<code>elemente</code>	INTEGER	Anzahl der Elemente bei Listefeldern Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation, da aktuell keine Listfelder verwendet werden.
<code>fkFeld</code>	INTEGER	Fremdschlüssel zu dem Teildatensatz und zu dem Feld, bilden zusammen die identifizierenden Merkmale
<code>fkBogen</code>	INTEGER	
<code>fkMussKann</code>	TEXT ( 1 )	M oder K, Unterscheidung zwischen Muss- und Kann-Feldern

<sup>40</sup> Gegebenenfalls im Kontext der Überschriften (Abschnitt B 2.2.4).

Feldname	Datentyp	Bemerkung
min	DOUBLE	Harte Untergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
max	DOUBLE	Harte Obergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
minWeich	DOUBLE	Weiche Untergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
maxWeich	DOUBLE	Weiche Obergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
ahinweis	TEXT ( 32 )	Name des HTML-Ausfüllhinweises ohne Endung .htm (Abschnitt B 2.2.5)

### Muss- und Kann-Felder

Jedes Bogenfeld ist als Muss- oder Kann-Feld zu deklarieren:

- Ein Muss-Feld (M) muss innerhalb eines angelegten Teildatensatzes immer ausgefüllt sein (Abschnitt B 2.2.2).<sup>41</sup>
- Kann-Felder (K) sind optionale Felder.
- Abhängige Muss-Felder (K) müssen nur unter bestimmten Bedingungen ausgefüllt werden. Wenn also logische Sachverhalte dem Ausfüllen von Kann-Feldern entgegenstehen, so dürfen sie nicht ausgefüllt werden. Diese Felder unterliegen Feldgruppenregeln und verfügen wie optionale Felder über den Attributwert K.

### Anzahl der Elemente von Listenfeldern

Das Attribut `elemente` ist nur relevant bei von Listenfeldern (vgl. Attribut `istListe` der Tabelle `Feld`) abgeleiteten Bogenfeldern (Bogenfeldlisten). Es gibt die Größe der Bogenfeldliste an. Wenn für eine Bogenfeldliste das Attribut `elemente` leer ist, so ist die Größe per Definition 1.

Wenn ein Listenfeld als Muss-Feld deklariert ist, so ist nur das erste Exportfeld der Liste ein Muss-Feld, die restlichen Elemente sind Kann-Felder. Wenn ein Listenfeld als Kann-Feld deklariert ist, so sind alle weiteren exportierten Elemente ebenfalls Kann-Felder.

### Felder – ein erster Schritt zur Prozess- und Datenintegration

Die Tabelle `Feld` (Tabelle 24) erleichtert dem Softwarehersteller den Abgleich seines Datenmodells mit dem Datenmodell des IQTIG. Gleiche Informationen in der Menge aller Dokumentationsbögen müssen dadurch nicht redundant abgebildet werden.

<sup>41</sup> In jedem Muss-Feld muss für jeden angelegten Teildatensatz einmal eine Angabe erfolgen.

Beispielsweise taucht das Feld `PRODUKTRISIKO` (Risikoeinstufung steriler Medizinprodukte) in den Modulen `NWIEA` und `NWIES` auf. Um die mehrfache Pflege dieser Felder zu vermeiden, wird ein Feld mit dem Namen `PRODUKTRISIKO` definiert und jeweils nur noch in der Tabelle `BogenFeld` referenziert.

Jedem Feld ist zwingend ein Basistyp zugeordnet (Abschnitt B 2.2.1). Bei Schlüsselfeldern muss auch ein Schlüssel assoziiert sein. Im Gegensatz zu den (technischen) Basistypen enthalten die Felder die medizinisch-fachliche Information der Datenfelder. Der fachliche Inhalt wird durch den Text im Attribut `bezeichnung`<sup>42</sup> beschrieben.

Identifizierendes Attribut eines Felds ist allein sein technischer Name (Attribut `name`). Dies ist wichtig für die Eindeutigkeit von Feldnamen innerhalb eines Moduls: Felder mit unterschiedlichen Typen oder unterschiedlichen Schlüsseln müssen unterschiedliche Namen haben.

Ein Feld kann als Skalar oder als Liste definiert sein. Diese Eigenschaft wird über das Attribut `istListe` gesteuert. Jedes von einem Listenfeld abgeleitete Bogenfeld ist automatisch eine Liste.<sup>43</sup> Die Anzahl der Elemente des von einem Feld abgeleiteten Bogenfelds wird über das Attribut `elemente` der Tabelle `BogenFeld` gesteuert.

Insbesondere für die Verwendung der richtigen Operatoren in den Plausibilitätsregeln und Feldgruppen ist die Listendefinition eines Felds wichtig.

Grundsätzlich gilt: Die Festlegung, ob ein Bogenfeld ein Skalar oder Listenfeld ist, wird durch die Tabelle `Feld` vorgegeben. Alle von einem Listenfeld abgeleiteten Bogenfelder sind automatisch auch Listenfelder. Die Größe der Liste wird individuell in der Tabelle `BogenFeld` konfiguriert.

Die Tabelle `Feld` bietet über die „Bogensicht“ hinausgehende Informationen.

Tabelle 24: Struktur der Tabelle `Feld`

Feldname	Datentyp	Bemerkung
<code>idFeld</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name
<code>bezeichnung</code>	TEXT	(Erlaubte Zeichen: A-Z, 0-9, Ziffer nicht am Anfang) Beschreibender Text auf dem Dokumentationsbogen (Standardwert für gleichnamiges Feld in Tabelle <code>BogenFeld</code> )
<code>laenge</code>	INTEGER	Anzahl der Zeichen in der Feldeingabemaske, enthält beim Typ <code>ZAHL</code> auch das Komma, bei <code>SCHLUESSEL</code> die Trennzeichen
<code>einheit</code>	TEXT(50)	Einheit des Felds (z.B. mm, Stunden)

<sup>42</sup> Das Attribut `bezeichnung` ist ein Standardtext für das gleichnamige Attribut der Tabelle `BogenFeld`. Im Eingabeformular wird die Bezeichnung aus der Tabelle `BogenFeld` angezeigt.

<sup>43</sup> Man beachte die Besonderheiten der Listenfelder beim Datenexport und in der Syntax der Plausibilitätsregeln.

Feldname	Datentyp	Bemerkung
formatAnweisung	TEXT	Regulärer Ausdruck für die Formatprüfung (z. B. [0-9]{9})
nachKommaLaenge	INTEGER	Anzahl der Nachkommastellen in der Feldeingabemaske (muss kleiner als laenge sein)
min	DOUBLE	Harte Untergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
max	DOUBLE	Harte Obergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
minWeich	DOUBLE	Weiche Untergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
maxWeich	DOUBLE	Weiche Obergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
istListe	BOOLEAN	Wenn istListe = WAHR, so sind die vom betreffenden Feld abgeleiteten Bogenfelder Listenfelder.
fkBasisTyp	INTEGER	Fremdschlüssel zur Tabelle Basistypen
fkSchluessel	INTEGER	Fremdschlüssel zur Tabelle Schlüsseltypen
fkKombiFeld	INTEGER	Optionaler Fremdschlüssel auf ein anderes Feld, welches Kombinationsfelder kennzeichnet

### Kombinationsfelder

Für manche Bogenfelder ist zwingend vorgeschrieben, dass sie innerhalb eines Moduls in Kombination mit einem anderen Bogenfeld existieren. Die Definition von Kombinationsfeldern geschieht mithilfe des optionalen Fremdschlüssels `fkKombiFeld` in der Tabelle `Feld`.

### Basistypen

Das Hauptmerkmal eines Basistyps ist der technische Typ eines Eingabefelds (z. B. Zeichenkette, numerischer Typ, Datum usw.). Wichtiges Charakteristikum ist die Beschreibung des Eingabeformats. Die Basistypen sind Voraussetzung für die Beschreibung einer formalen Regelsyntax (Abschnitt B 2.3.2).

Das identifizierende Merkmal eines Basistyps ist sein technischer Name (Attribut `name`).

Tabelle 25: Struktur der Tabelle *Basistyp*

Feldname	Datentyp	Bemerkung
idBasistyp	INTEGER	Primärschlüssel
name	TEXT	Technischer Name (muss eindeutig sein)
standardtyp	TEXT	Entsprechender Standarddatentyp
bezeichnung	TEXT	Beschreibender Text
format	TEXT	Formatdefinition, z.B. TT.MM.JJJJ beim Basistyp Datum
formatRegExp	TEXT	Regulärer Ausdruck für die Formatprüfung
stdLaenge	INTEGER	Vorschlagsfeld für das gleichnamige Feld in der Tabelle <code>Feld</code> (einschließlich Vorzeichen und Komma)
stdNachKommaLaenge	INTEGER	Vorschlagsfeld für das gleichnamige Feld in der Tabelle <code>Feld</code>

**Hinweis**

- In Zeichenketten (Basistyp TEXT) sind alle Zeichen des ASCII-Formats mit einem Kode  $\geq 32$  erlaubt. Ausgenommen sind das Semikolon, die doppelten Anführungsstriche und Hochkommata.
- Es gibt zwei Arten von Schlüssel: numerisch und nichtnumerisch
- Das Komma trennt die Nachkommastellen, Vorzeichen + und – sind erlaubt.

**Schlüssel**

Identifizierendes Merkmal eines Schlüssels (Kodesystem) ist sein technischer Name. Die meisten Schlüsselcodes sind in der Tabelle `Schlüsselwert` definiert.

Tabelle 26: Struktur der Tabelle *Schluesse1*

Feldname	Datentyp	Bemerkung
idSchluesse1	INTEGER	Primärschlüssel
name	TEXT	Technischer Name (muss eindeutig sein)
bezeichnung	TEXT	Beschreibender Text
extern	BOOLEAN	Zeigt an, ob der Schlüssel in der Tabelle <code>Schluesse1</code> (= FALSCH) oder in einer externen Tabelle gespeichert (= WAHR) ist.
externVerweis	TEXT	Verweis auf die Quelle des externen Schlüssels

Feldname	Datentyp	Bemerkung
zahl	BOOLEAN	Wenn WAHR, sind die Werte im Attribut <code>code</code> der zugehörigen Schlüsselwerte als ganze Zahl kodiert, ansonsten als Zeichenkette.
sortierNrVerwendet	BOOLEAN	Flag, das anzeigt, ob für die Reihenfolge das Attribut <code>sortierNr</code> der Tabelle <code>Schlüsselwert</code> herangezogen wird.
fkMutterSchluessel	INTEGER	Referenz auf einen übergeordneten Schlüssel.

Schlüsselcodes können auf zwei Arten interpretiert werden: Wenn das Attribut `zahl` gesetzt ist, so werden die Codes als ganze Zahl gedeutet, ansonsten werden sie als Zeichenketten interpretiert. In der Syntax der Plausibilitätsregeln werden die letztgenannten Codes in einfache Hochkommata gesetzt (Abschnitt B 2.3.2).

### Beispiel:

Attribut **zahl** bei Schlüsselfeldern

- Felder des Basistyps `NUMSCHLUESSEL` haben das Attribut `zahl = WAHR`.
- Felder des Basistyps `SCHLUESSEL` haben das Attribut `zahl = FALSCH`. Es handelt sich um alphanumerische Schlüssel, die Buchstaben, Ziffern oder Sonderzeichen verwenden (z.B. `YPN0`). Hierbei kann es sich auch um Werte handeln, die lediglich Ziffern verwenden, aber mit führender Null beginnen (z.B. `01`).

### Externe Schlüsselkataloge

Externe Schlüsselkataloge sind über das Attribut `extern` deklariert. Hinweise zu den Bezugsquellen sind in der Spalte `externVerweis` zu finden (z.B. [www.dimdi.de](http://www.dimdi.de)). Diese externen Schlüsselkataloge werden nicht vom IQTIG bereitgestellt und somit auch nicht verantwortet.

Hinweise zu den Bezugsquellen sind in der Spalte `externVerweis` zu finden. Ein Verweis auf eine Bezugsquelle kann unabhängig vom Attribut `extern` angegeben werden.

Die Schlüsselcodes, die dem IQTIG zum Zeitpunkt der Publikation der QS-Spezifikation bekannt sind, sind in der Tabelle `Schlüsselwert` enthalten. Spätere Schlüsseländerungen bzw. Fortschreibungen müssen (z.B. vom Softwareanbieter und von der datenentgegennehmenden Stelle) selbstständig und zeitnah über die jeweilige Bezugsquelle bezogen werden.

Die Schlüsselcodes sind in der Tabelle `Schlüsselwert` enthalten. Spätere Schlüsseländerungen bzw. -fortschreibungen werden vom IQTIG zeitnah übernommen.

## Schlüsselwerte

Tabelle 27 gibt einen Überblick über die Datenbanktabelle `SchlüsselWert`, in der die Codes und Bezeichnungen der Schlüssel hinterlegt sind. Identifizierendes Merkmal ist hier eine Kombination der Spalten `fkSchlüssel` und `code`. Das bedeutet, dass jeder Schlüsselcode innerhalb eines Schlüssels nur einmal vorkommen darf.

Tabelle 27: Struktur der Tabelle `SchlüsselWert`

Feldname	Datentyp	Bemerkung
<code>idSchlüsselWert</code>	INTEGER	Primärschlüssel
<code>fkSchlüssel</code>	INTEGER	Fremdschlüssel zur Tabelle <code>Schlüssel</code>
<code>code</code>	TEXT ( 50 )	Schlüsselcode (entweder numerisch oder alphanumerisch kodiert)
<code>bezeichnung</code>	TEXT	Textliche Definition des Schlüsselwertes
<code>sortierNr</code>	INTEGER	Optionale Angabe zur Reihenfolge der Schlüsselwerte: Wenn belegt, so ist diese Reihenfolge bei der Anzeige in der Erfassungssoftware einzuhalten.

Das Attribut `code` der Tabelle `SchlüsselWert` ist ein Textfeld, das in Abhängigkeit vom Wert des Attributes `zahl` im zugeordneten Schlüssel entweder numerisch oder nichtnumerisch interpretiert wird. Wenn in einer Plausibilitätsregel (Abschnitt B 2.3.2 und B 2.3.7) Felder mit numerischen Schlüsseln (Basistyp `NUMSCHLUESSEL`) vorkommen, so werden bei der Evaluierung der Regel die Schlüsselcodes wie ganze Zahlen behandelt.

### Sortierung der Codes

- Für die Codes (Attribut `SchlüsselWert.code`) eines Schlüssels ist eine Sortierung definiert. Die Art der Sortierung wird über die Attribute `zahl` und `sortierNrVerwendet` der Tabelle `Schlüssel` festgelegt.
- Numerische Sortierung: Wenn `sortierNrVerwendet = FALSCH` und `zahl = WAHR`, so sind die Codes nach der Spalte `code` der Tabelle `Schlüssel` numerisch zu sortieren.
- Alphanumerische Sortierung: Wenn `sortierNrVerwendet = FALSCH` und `zahl = FALSCH`, so sind die Codes nach der Spalte `code` der Tabelle `Schlüssel` alphanumerisch zu sortieren.
- Spezielle Sortierung: Wenn `sortierNrVerwendet = WAHR`, so sind die Codes nach den Werten in der Spalte `sortierNr` der Tabelle `Schlüssel` numerisch zu sortieren.

### Suchfunktion bei Schlüsseln mit einer großen Anzahl von Codes

Bei Schlüsseln mit einer großen Anzahl von Codes soll eine anwenderfreundliche Möglichkeit zur Auswahl der passenden Codes bereitgestellt werden. Die Umsetzung als Auswahlliste (z.B. Combobox) führt zu erhöhtem Dokumentationsaufwand, falls der Anwender über Pfeiltasten oder Schiebepfeile zum passenden Code navigieren muss. Ergänzend soll daher eine Suchfunktion

realisiert werden, die eine Suche über die Attribute `SchlüsselWert.code` oder `SchlüsselWert.bezeichnung` ermöglicht. Die zu realisierenden Anwendungsfälle werden in den folgenden Beispielen erläutert.

#### 2.2.4 Überschriften

Die Überschriften der Dokumentationsbögen in der Spezifikation sind in der Tabelle Abschnitt zu finden.

Tabelle 28: Struktur der Tabelle *Abschnitt*

Feldname	Datentyp	Bemerkung
<code>idAbschnitt</code>	INTEGER	Primärschlüssel
<code>bezeichnung</code>	TEXT	Text der Überschrift
<code>ebene</code>	INTEGER	Zeigt die Hierarchie der Überschriften an
<code>fkStartBogenFeld</code>	INTEGER	Fremdschlüssel auf das erste zur Überschrift gehörende Bogenfeld
<code>fkEndeBogenFeld</code>	INTEGER	Fremdschlüssel auf das letzte zur Überschrift gehörende Bogenfeld

Zu jeder Überschrift ist angegeben, bei welchem Bogenfeld sie beginnt und bei welchem Bogenfeld sie endet. Über das Attribut `ebene` lassen sich auch Teilüberschriften realisieren. Ein Bogenfeld kann somit mehreren Überschriften zugeordnet sein.



#### **Achtung**

Die in der Spezifikationsdatenbank hinterlegten Überschriften sind in die Eingabemasken (z. B. der QS-Dokumentationssoftware) zu integrieren. Viele Datenfelder sind für den Anwender erst im Kontext der Überschriften verständlich.

#### 2.2.5 Ausfüllhinweise

Die Ausfüllhinweise zu den Datenfeldern sind in einem separaten ZIP-Archiv enthalten, das nach dem Benennungsschema für Spezifikationskomponenten bezeichnet wird (Einleitung, Abschnitt 1.1). Jeder Ausfüllhinweis ist ein HTML-Dokument.

In der Spalte `ahinweis` der Tabelle `BogenFeld` ist festgelegt, welcher HTML-Ausfüllhinweis mit einem Datenfeld verknüpft ist:

`<aHinweis>.htm` = Name der HTML-Datei

Wenn der Eintrag in `ahinweis` leer ist, so existiert für das betreffende Bogenfeld kein Ausfüllhinweis. Das Attribut `fkAhinweisTyp` lässt die Differenzierung drei verschiedener Arten von Ausfüllhinweisen zu:

Tabelle 29: Arten von Hinweistypen

fkAhinweisTyp	Beschreibung	Beispiel
Feldbezogen	Der Ausfüllhinweis bezieht sich auf den entsprechenden Datensatz in der Tabelle <code>Feld</code> . Der Ausfüllhinweis ist modulunabhängig.	<code>IKNRKH.htm</code> Der Ausfüllhinweis bezieht sich auf verschiedene Module, beispielsweise auf <code>NWIEA</code> und <code>NWIES</code> .
Modulspezifisch	Soll sich ein Ausfüllhinweis nur auf ein bestimmtes Modul beziehen, kann der Attributwert <code>modulspezifisch</code> ausgewählt werden.	<code>LLPROPH\$NWIEA.htm</code> Der Ausfüllhinweis bezieht sich nur auf das Modul <code>NWIEA</code> .
Speziell	Soll es für verschiedene Datenfelder der Tabelle <code>Feld</code> einen gemeinsamen Ausfüllhinweis geben, kann dieser als speziell deklariert werden. Der Attributwert <code>ahinweis</code> definiert den Namen des Ausfüllhinweises.	<code>DESINFECTINTSTAT.htm</code> Die Felder <code>DESINFEKTIONOP</code> und <code>DESINFEKTION</code> haben denselben Ausfüllhinweis.

Die Zuordnung von Bogenfeldern und Ausfüllhinweisen ist auch in der Abfrage `Ausfüllhinweise` dargestellt. Sie zeigt Modul/Teildatensatz, Zeile, Feldname, Bezeichnung und den HTML-Dateinamen des Ausfüllhinweises zu dem Bogenfeld. Im Gegensatz zur Tabelle `Bogenfeld` ist hier die Endung `.htm` mit angegeben.

## 2.3 Plausibilitätsprüfungen

Es wird zwischen zwei Arten von Plausibilitätsprüfungen unterschieden, die in `TabelleRegelTyp` definiert sind:

- harte Prüfungen
- weiche Prüfungen in der QS-Dokumentationssoftware

Tabelle 30: Tabelle `RegelTyp`

idRegelTyp	bezeichnung
H	hart
W	weich

Weiterhin wird zwischen sogenannten **Einzelregeln** (Abschnitt B 2.3.4) und **Feldgruppenregeln** (Abschnitt B 2.3.7) unterschieden.

### 2.3.1 Die Regeltabelle

Die Bedingungen für un plausible Angaben<sup>44</sup> sind in der Tabelle `Regeln` abgelegt. Die hier beschriebenen Prüfungen sind in der Spezifikationsdatenbank für QS-Dokumentation hinterlegt. Die Syntax ist in Abschnitt B 2.3.4 beschrieben. Die Bedingungen sind möglichst kurz gefasst (Vermeidung von durch `ODER` verknüpften Teilbedingungen). Jede Bedingung kommt nur einmal innerhalb eines Moduls vor.

Tabelle 31: Struktur der Tabelle `Regeln`

Feldname	Datentyp	Bemerkung
<code>idRegeln</code>	INTEGER	Primärschlüssel
<code>fkModul</code>	INTEGER	Fremdschlüssel zur Tabelle <code>Modul</code>
<code>bedingung</code>	MEMO	Entsprechend der Syntax definierte Regeln
<code>meldung</code>	MEMO	Fehlermeldung: Diese Texte sind bei Regeln mit Bezug zu Feldgruppen automatisch generiert.
<code>alternativmeldung</code>	MEMO	Alternative Fehlermeldung: Wenn hier ein Text vorhanden ist, so ist dieser anstelle des Textes in der Spalte <code>meldung</code> zu verwenden.
<code>fkMehrfachRegel</code>	INTEGER	Fremdschlüssel zur Tabelle mit mehrfach vorkommenden Regeln, die mithilfe von Ersatzbedingungen nach dem Export gültig sind. Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
<code>fkFeldGruppe</code>	INTEGER	Optionaler Fremdschlüssel zur Tabelle <code>FeldGruppe</code> : Indikator dafür, ob eine Regel aus einer Feldgruppe generiert wurde.
<code>fkRegelTyp</code>	TEXT(1)	Fremdschlüssel zur Tabelle <code>RegelTyp</code> : Die Regeltypen sind die in Abschnitt A 2.3 beschriebenen Arten der Plausibilitätsprüfungen: H, W oder D
<code>gueltigNachExport</code>	BOOLEAN	Regeln, die den Wert <code>FALSCH</code> haben, können von Datenannahmestellen nicht evaluiert werden. Stattdessen werden die referenzierten Ersatzbedingungen der Tabelle <code>MehrfachRegel</code> evaluiert (falls definiert).

<sup>44</sup> Eine Plausibilitätsregel müsste eigentlich „Unplausibilitätsregel“ heißen, weil sie un plausible Zustände beschreibt, die zu Fehlermeldungen führen.

### Regelfelder (Bogenfelder einer Regel)

Die Tabelle `RegelFelder` (Tabelle 32) ist eine Verknüpfungstabelle zwischen den Tabellen `Regeln` und `BogenFeld`. Durch gezielte Abfragen erhält man unter Verwendung dieser Tabelle einen Überblick über Folgendes:

- Bogenfelder, die in einer Regel verwendet werden.
- Regeln, die sich auf ein Bogenfeld beziehen.

Tabelle 32: Struktur der Tabelle `RegelFelder`

Feldname	Datentyp	Bemerkung
<code>fkBogenFeld</code>	INTEGER	Fremdschlüssel zu den Tabellen <code>Feld</code> und <code>Regeln</code> , bilden zusammen den Primärschlüssel
<code>fkRegeln</code>	INTEGER	

### Mehrfachregeln (Ersatzregeln zur Prüfung nach dem Export)

Wenn in einer Regel von der Pseudonymisierung betroffene Datenfelder benutzt werden, so kann diese von Datenannahmestellen nicht evaluiert werden. Stattdessen wird für solche Regeln in der Tabelle `MehrfachRegel` eine Ersatzbedingung definiert, deren Referenzierung in der Tabelle `Regeln` definiert ist. Die Ersatzbedingung ist von den Datenannahmestellen zu evaluieren.

Für die Module `NWIES` und `NWIEA` sind derzeit keine Mehrfachregeln definiert. Weitere Erläuterungen zu Mehrfachregeln sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

### Weitere Regeln

Weitere feldübergreifende Regeln sind die in Abschnitt B 2.2.2 beschriebenen Existenzbedingungen für das Anlegen von abhängigen Teildatensätzen (Attribut `existenzBedingung` in Tabelle `Bogen`).

#### 2.3.2 Regelsyntax

Bedingungen sind in den Tabellen `Regeln`, `MehrfachRegel` und `Bogen` definiert. Die den Bedingungen zugrunde liegende Regelsyntax wird in diesem Abschnitt beschrieben. Jede Regel ist ein logischer Ausdruck, dessen Ergebnis `WAHR` oder `FALSCH` lautet. Jede Regel bezieht sich auf einen eingegebenen Datensatz eines Moduls, dessen Daten in Variablen gespeichert sind.

Die Regelsyntax lehnt sich an die logischen Ausdrücke in bekannten Programmiersprachen an. Jedoch haben die Operatoren deutsche Namen, z.B. `UND` statt `AND` oder `ODER` statt `OR`. Die Regelsyntax ist als Pseudocode zu verstehen.

### Typen

Die möglichen Typen der Datenfelder sind in Tabelle 33 aufgelistet.

Tabelle 33: Basistypen der Datenfelder in den Plausibilitätsregeln

Basistyp	Bezeichnung	Beispiele (Literele)
BOOL	Boolesche Variable	WAHR, FALSCH
TEXT	Zeichenkette (String)	"Spezifikation"
GANZEZAHL <sup>45</sup>	... -2, -1, 0, 1, 2, 3, ...	1
ZAHL	Zahl (mit oder ohne Nachkommastellen)	25,4 oder -100,8
DATUM	Zehnstelliges Datum	'01.01.2012'
MONDATUM	Monatsdatum	'04.2012'
QUARTDATUM	Quartalsdatum	'3/2012'
JAHRDATUM	Jahresdatum	2012
NUMSCHLUESSEL	Numerisch kodierter Schlüssel (wie GANZEZAHL)	1
SCHLUESSEL	Alphanumerischer Schlüssel	'19.1', '07'
UHRZEIT	Uhrzeit	'10:15'

In der Spezifikation für die QS-Dokumentation wird zwischen NUMSCHLUESSEL und SCHLUESSEL unterschieden:

- Schlüsselwerte verfügen über den Datentyp NUMSCHLUESSEL, wenn es sich bei den Codes um ganze Zahlen handelt. Da diese ein numerischer Schlüssel ist, darf er nicht in Hochkommata gesetzt werden.
- Schlüsselwerte, die alphanumerische Codes beinhalten, haben den Basistyp SCHLUESSEL. Die OPS-Schlüssel (z.B. '5-282.0') und die ICD-10-GM-Schlüssel verfügen über diesen Datentyp, der in Hochkommata geschrieben wird.



#### Achtung

Datumsangaben (Datum, Monats-, Quartalsdatum) müssen in Hochkommata gesetzt werden. Eine Ausnahme ist das Jahresdatum (JAHRDATUM), da es sich hierbei um eine ganze Zahl handelt.

#### Felder

Feldnamen bestehen aus maximal 32 Zeichen und dürfen nur die Buchstaben A bis Z (Großbuchstaben) und die Ziffern 0 bis 9 enthalten. Ein Feldname muss immer mit einem Buchstaben beginnen. Umlaute und Sonderzeichen sind in Feldnamen nicht erlaubt. Ein Feldname darf kein reserviertes Wort sein (z.B. LEER).

<sup>45</sup> Beim Typ GANZEZAHL sind auch negative ganze Zahlen erlaubt.



### Achtung

In einer Regel dürfen nur die Feldnamen der im betreffenden Modul definierten Bogenfelder<sup>46</sup> enthalten sein. Bei der Evaluierung von Regeln werden die aktuellen Werte der referenzierten Bogenfelder eingesetzt. Kann-Bogenfelder können auch un- ausgefüllt sein, also den Wert LEER haben.

### Listenfelder

Ein Bogenfeld wird dann als Liste interpretiert, wenn im referenzierten Feld (Tabelle `Feld`) der Wert des Attributs `Feld.istListe = WAHR` ist. Andernfalls ist das Bogenfeld ein Skalar. Bei der Formulierung von Regeln ist darauf zu achten, dass Listenfelder nicht bei jedem Operator als Operand fungieren können. Listenfelder dürfen z.B. nicht voneinander subtrahiert werden.

### Literale

Alphanumerische Literale (z.B. `SCHLUESSEL`) werden von einfachen Hochkommata eingeschlossen, während Zeichenketten vom Datentyp `TEXT` in Anführungszeichen gesetzt werden müssen.<sup>47</sup>

Dies gilt nicht für numerische Literale (`GANZEZAHL`, `ZAHL`, `NUMSCHLUESSEL` und `JAHR-DATUM`) und Literale des Datentyps `BOOL` (Wahrheitswerte).

### Listen von Literalen

Literale können sowohl als Skalare als auch als Listen angesprochen werden. Der Separator einer Liste von Literalen ist das Semikolon. Um zu prüfen, ob alle Listenfelder ausgefüllt sind, wird die Liste über den Wert `LEER` angesprochen.

Außerdem gibt es Teildatensatz-Listenfelder, die im Abschnitt B 2.3.6 beschrieben werden.

### Operatoren

Tabelle 34 gibt einen Überblick über die in der Syntax zulässigen Operatoren. Der aktuelle Überblick über alle zulässigen Operationen (inkl. Operanden) ist in Tabelle `SyntaxOperator` in der QSDOK-Datenbank zu finden.

In Tabelle 34 hat jeder einzelne Operator eine Präzedenzstufe (höchste Präzedenzstufe ist 0). Operatoren, die die gleiche Stufe haben, werden nach den Regeln der Assoziativität aufgelöst.

Tabelle 34: Präzedenz und Assoziativität der Operatoren

Präzedenz	Assoziativität	Operator	Erläuterung
0	links	IN	Operator zum Vergleich einer Variablen mit einer Liste (z.B. ein Datenfeld mit Schlüsselwerten). Die Variable und die Feldelemente müssen gleichen Typs sein.
	links	NICHTIN	
	links	EINSIN	

<sup>46</sup> Bei den Ersatzregeln in Tabelle `MehrFachRegel` sind stattdessen die Exportfelder des Moduls erlaubt.

<sup>47</sup> Beim Export entfallen die begrenzenden Zeichen.

Präzedenz	Assoziativität	Operator	Erläuterung
	links	JEDESIN	Operator zum Vergleich einer Liste mit einer anderen Liste oder einem Listenelement (z.B. ein Listenfeld mit einem Schlüsselwert oder ein Listenfeld mit einer OPS-Liste). Die Listenelemente müssen gleichen Typs sein.
	links	EINSNICHTIN	
	links	KEINSIN	
1	links	*	Operator für die Multiplikation „mal“
	links	/	Operator für die Division „geteilt“
2	links	+	Operator für die Addition „plus“
	links	-	Operator für die Subtraktion „minus“
3	links	<	Vergleichsoperator „kleiner“
	links	>	Vergleichsoperator „größer“
	links	<=	Vergleichsoperator „kleiner gleich“
	links	>=	Vergleichsoperator „größer gleich“
4	links	=	Vergleichsoperator „gleich“
	links	<>	Vergleichsoperator „ungleich“
5	rechts	NICHT	Logischer Operator: „NICHT“
6	links	UND	Logischer Operator: „UND“
7	links	ODER	Logischer Operator: „ODER“

### Prüfung auf LEER mit Vergleichsoperatoren

Die Prüfung auf LEER von in Regeln verwendeten Kann-Feldern, welche an anderer Stelle in der Regel mit einem anderen Operator als <> oder = geprüft werden sollen, findet auf der linken Seite einer ODER-Verknüpfung statt. Hintergrund dieser Syntaxregel ist, dass die Vermeidung von Laufzeitfehlern bei der Evaluation ermöglicht werden soll.

#### Beispiel:

FELD = LEER ODER FELD OPERATOR OPERAND

Beispielsweise kann bei leeren Feldwerten und der vorgegebenen Linksassoziativität des ODER-Operators die Evaluation bei leerem Feldwert vor der Evaluation des rechtsstehenden Ausdrucks mit der Rückgabe von WAHR abgebrochen werden. Ein Laufzeitfehler, der sich z.B. bei einem Vergleich von LEER < WERT ergeben würde, kann so nicht entstehen.

### Operatoren zum Vergleich einer Variablen mit einer Liste

Folgende Operatoren erfordern entweder nur rechts oder links und rechts Listenfelder:

- nur rechts: IN, NICHTIN

- links und rechts: EINSIN, KEINSIN, JEDESIN, EINSNICHTIN

Operatoren mit beidseitigen Listenfeldern als Operanden:

- EINSIN: Wenn mindestens ein Element aus der linken Liste in der rechten Liste enthalten ist, so ist der Ausdruck wahr (nichtleere Schnittmenge).
- KEINSIN: Wenn kein Element der linken Liste in der rechten Liste enthalten ist, so ist der Ausdruck wahr (leere Schnittmenge). Dieser Operator ist redundant, da er auch durch Negation des EINSIN-Operators abgedeckt ist.
- JEDESIN: Der Ausdruck ist dann wahr, wenn jedes Element der linken Liste in der rechten Liste enthalten ist (Teilmenge).
- EINSNICHTIN: Der Ausdruck ist dann wahr, wenn mindestens ein Element der linken Liste nicht in der rechten Liste enthalten ist (nichtleere Differenz).

---

#### Beispiel:

- Die Operation `GANZEZAHL := DATUM1 - DATUM2` liefert als Ergebnis die Differenz zwischen zwei Kalenderdaten in Tagen.
- Die Operation `ZAHL := UHRZEIT1 - UHRZEIT2` liefert als Ergebnis die Differenz zwischen zwei Uhrzeiten in Minuten.

---

Eine Besonderheit bei Listenoperationen ist die Prüfung, ob alle Elemente einer Liste ausgefüllt sind:

---

#### Beispiel:

`NICHT FELD JEDESIN (LEER)`

Diese Bedingung erfordert, dass zumindest ein Listenelement ausgefüllt ist. Beispielsweise erfüllt `FELD := (WERT;LEER;LEER;LEER)` die Bedingung.

Gleichwertig ist die Regel:

`FELD EINSNICHTIN (LEER)`

---

Folgende Operatoren sind komplementär:

IN und NICHTIN

EINSIN und KEINSIN

JEDESIN und EINSNICHTIN

Folgende Ausdrücke sind gleich:

A EINSNICHTIN B

NICHT A JEDESIN B

### 2.3.3 Funktionen

Eine Funktion ist gekennzeichnet durch ihren Namen, an den sich unmittelbar (ohne Leerzeichen) ein Listenausdruck anschließt. Funktionen ohne Übergabeparameter werden ähnlich wie in C oder Java durch ein Klammerpaar abgeschlossen. Funktionen können nicht nur in Regeln, sondern auch zur Berechnung von Exportfeldern genutzt werden (Abschnitt B 2.4.3).

Der aktuelle Stand der in der Syntax verwendeten Funktionen ist in der Tabelle `SyntaxFunktion` der Spezifikation zu finden. In der QSDOK-Spezifikationsdatenbank der einrichtungsbezogenen QS-Dokumentation sind derzeit keine Funktionen hinterlegt.

### 2.3.4 Syntaxvariablen

Der Eingangsdatensatz bildet die medizinische Routinedokumentation ab, die in jedem Krankenhausinformationssystem (KIS) enthalten ist. Syntaxvariablen in der QS-Dokumentation dienen der technischen Darstellung der automatischen Generierung von Angaben aus dem Eingangsdatensatz. Diese sind in Tabelle `SyntaxVariable` hinterlegt. In den Modulen `NWIES` und `NWIEA` werden derzeit keine Syntaxvariablen verwendet. Weiterführende Erläuterungen zu Syntaxvariablen sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

### 2.3.5 Einzelregeln

Sogenannte Einzelregeln können sich als feldbezogene Prüfungen auf ein einziges Datenfeld oder als feldübergreifende Prüfungen auf mehrere Datenfelder beziehen. Einzelregeln sind von den in Abschnitt B 2.3.7 beschriebenen Feldgruppen zu unterscheiden.

**Feldbezogene Prüfungen** – beispielsweise Wertebereichsüberprüfungen – sind in der formalen Regelsyntax in Tabelle **Regeln** formuliert.

Unter feldbezogenen Prüfungen sind aber auch die in Abschnitt B 2.3.8 beschriebenen Prüfungen des Formates, der Feldlänge, der Wertebereiche, Prüfungen von Schlüsselkodes und von Muss-Feldern zu verstehen. Für diese Prüfungen gibt es keine formale Regelsyntax in Tabelle **Regeln**.

#### **Feldübergreifende Regeln**

- haben eine eigene Syntax,
- haben geringe Komplexität,
- haben einfache, dem Anwender verständliche Fehlertexte,
- enthalten alle Teilregeln der Feldgruppen,
- haben gewöhnlich den Bezug zu zwei oder mehreren Feldern,
- können zum Teil direkt nach der Benutzereingabe in ein Feld geprüft werden,
- enthalten Bedingungen für un plausible Angaben<sup>48</sup>.

---

<sup>48</sup> Eine Plausibilitätsregel müsste eigentlich „Unplausibilitätsregel“ heißen, weil sie un plausible Zustände beschreibt, die zu Fehlermeldungen führen.

Feldübergreifende Regeln können auch teildatensatzübergreifende Regeln sein, wenn die Datenfelder der Regel aus mehreren Teildatensätzen eines Moduls stammen (Abschnitt B 2.3.6).

### 2.3.6 Teildatensatzübergreifende Regeln

Eine Regel ist teildatensatzübergreifend, wenn die Datenfelder der Regel aus mehreren Teildatensätzen eines Moduls stammen.

Es gibt zwei Arten von teildatensatzübergreifenden Regeln:

1. Die Felder sind in verschiedenen Teildatensätzen eines Moduls definiert.
2. Ein Feld der Regel ist in einem wiederholbaren Teildatensatz definiert und die Regel bezieht sich auf alle Werte des Datenfeldes innerhalb eines Datensatzes (= Summe aller Teildatensätze eines Vorgangs).

### Regeln mit Teildatensatz-Listenfeldern

Zu jedem skalaren Datenfeld eines wiederholbaren Teildatensatzes existiert ein Teildatensatz-Listenfeld (kurz TDS-Listenfeld), das über das @-Zeichen vor dem Feldnamen angesprochen wird. Das TDS-Listenfeld enthält sämtliche Werte des betreffenden Datenfeldes, die innerhalb der QS-Dokumentation eines Falles existieren.

### 2.3.7 Feldgruppenregeln

Logische Abhängigkeiten von Bogenfeldern werden über Feldgruppen dargestellt. Die Plausibilitätsregeln, die einen Bezug zu einer Feldgruppe aufweisen (Tabelle `Regeln`), werden anhand der Feldgruppendefinition (Tabelle `FeldgruppeFelder`) automatisch generiert. Die Menge der abgeleiteten Einzelregeln wird in diesem Abschnitt erläutert.

Die möglichen Antworten<sup>49</sup> eines jeden Datenfeldes werden in zwei Gruppen aufgeteilt. Die erste Gruppe ist die Menge der positiven, die zweite Gruppe die Menge der negativen Antworten.<sup>50</sup>

Typische positive Antworten sind beispielsweise:

```
Feld <> LEER oder Feld IN (2;3)
```

Die komplementären negativen Antworten würden entsprechend wie folgt lauten:

```
Feld = LEER oder Feld NICHTIN (2;3)
```

Eine Feldgruppe kann ein Filterfeld haben. Wenn die Antwort dieses Filterfeldes negativ ausfällt (Bspw. Bedingung: `Feld = 3`; Antwort: `Feld <> 3`), so darf keines der abhängigen Felder positiv beantwortet werden.

Tabelle 35 gibt einen Überblick über die Typen von Feldgruppen. Der aktuelle Stand findet sich in der Tabelle `FeldGruppenTyp` der Spezifikation.

<sup>49</sup> Die Antworten eines Datenfeldes umfassen hier neben möglichen Werten (z.B. Schlüsselwerten) oder Wertemengen auch die Kategorie „nicht ausgefüllt“ (LEER).

<sup>50</sup> Die negativen Antworten sind abhängig von der definierten Bedingung eines Feldes in der entsprechenden Feldgruppe.

Tabelle 35: Typen von Feldgruppen

Name	Bemerkung
<b>mit Filterfeld</b>	
EF_FILTER	Einfachauswahl, genau ein abhängiges Feld muss positiv beantwortet sein
EF_OPTIONAL_FILTER	Einfachauswahl, genau ein abhängiges Feld kann positiv beantwortet sein
MF_OPTIONAL_FILTER	Mehrfachauswahl, alle abhängigen Felder können positiv beantwortet sein
MF_MINDESTENS1_FILTER	Mehrfachauswahl, mindestens ein abhängiges Feld muss positiv beantwortet sein
MF_ALLES_FILTER	Mehrfachauswahl, alle abhängigen Felder müssen positiv beantwortet sein
<b>ohne Filterfeld</b>	
EF	Einfachauswahl, genau ein Feld muss positiv beantwortet sein
MF_OPTIONAL	Mehrfachauswahl, alle Felder können positiv beantwortet sein
MF_MINDESTENS1	Mehrfachauswahl, mindestens ein Feld muss positiv beantwortet sein
UND	Einfache Regel mit Und-Verknüpfungen

In der Tabelle `BogenFeld` sind abhängige Datenfelder einer Feldgruppe immer als Kann-Felder definiert. Nach Abhängigkeit der Feldgruppenlogik können/müssen diese Felder leer bleiben oder zwingend ausgefüllt werden. Im letztgenannten Fall können die Datenfelder auch als bedingte Muss-Felder bezeichnet werden.

Die Muss- oder Kann-Definition der Datenfelder (Bogen- und Ersatzfelder) im Exportformat unterliegt ebenfalls der Feldgruppenlogik. Ist die Berechnung eines Ersatzfeldes von bedingten Datenfeldern abhängig, so gilt die Feldgruppenlogik auch für diese Ersatzfelder. Wenn die bedingten Datenfelder zwingend ausgefüllt werden müssen, so muss auch das Ersatzfeld zwingend berechnet bzw. exportiert werden.

### Struktur der Tabellen `FeldGruppe` und `FeldgruppeFelder`

Die Feldgruppen sind in den Tabellen `FeldGruppe` und `FeldgruppeFelder` definiert. In der Tabelle `FeldGruppe` (Tabelle 36) sind Name, Typ und die Zuordnung zu einem Modul definiert. Die Verknüpfungstabelle `FeldgruppeFelder` (Tabelle 37) definiert die abhängigen Bogenfelder. Zusätzlich wird hier festgelegt, welche Bogenfelder der Feldgruppe als Filterfeld dienen.

Tabelle 36: Struktur der Tabelle *FeldGruppe*

Feldname	Datentyp	Bemerkung
idFeldGruppe	INTEGER	Primärschlüssel
name	TEXT ( 64 )	Technischer Name der Feldgruppe
fkModul	INTEGER	Obligatorischer Fremdschlüssel zu einem Modul
fkFeldgruppenTyp	INTEGER	Obligatorischer Fremdschlüssel zu einem Feldgruppentyp
hinweis	TEXT	Bei Filter-Feldgruppen relevant für die Gestaltung der Eingabemaske. Der Hinweistext informiert den Anwender über die Bedingungen, welche das Ausfüllen von ein oder mehreren abhängigen Feldern erforderlich machen. Der Hinweistext kann bei der Erstellung der Eingabemasken verwendet werden.
fkFilterFeldTyp	CHAR ( 1 )	Attribut wird bei Feldgruppen mit mehreren Filterfeldern gesetzt: O = Oder-Verknüpfung der positiven Filterbedingungen U = Und-Verknüpfung der positiven Filterbedingungen
fkRegelTyp	CHAR ( 1 )	Fremdschlüssel zur Tabelle <i>RegelTyp</i> : Die Regeltypen sind die in Abschnitt A 2.3 beschriebenen Arten der Plausibilitätsprüfungen: H, W Die generierten Einzelregeln der Feldgruppe haben den gleichen Regeltyp.
nurPositiv	BOOLEAN	Nur bei Filter-Feldgruppen wirksam: Wenn WAHR, dann umfasst die Feldgruppe nur diejenigen Regeln, welche sich auf die positive Filterbedingung beziehen.
grauWennNegativ	BOOLEAN	Definiert eine Layout-Feldgruppe, wenn WAHR (siehe unten S. 104, Layout-Feldgruppen)

Tabelle 37: Struktur der Tabelle *FeldgruppeFelder*

Feldname	Datentyp	Bemerkung
idFeldgruppeFelder	INTEGER	Primärschlüssel
fkFeldGruppe	INTEGER	Obligatorischer Fremdschlüssel zur Feldgruppe
fkBogenFeld	INTEGER	Obligatorischer Fremdschlüssel zum Bogenfeld

Feldname	Datentyp	Bemerkung
bedingung	TEXT	Positive Bedingung für das jeweilige Bogenfeld
istFilter	BOOL	Legt fest, ob das jeweilige Bogenfeld ein Filterfeld ist
bezeichnungSchluesselListe	TEXT	Abkürzende Bezeichnung für eine Schlüsseliste in der Bedingung, wird beim Generieren von Fehlermeldungen verwendet.
tdsFilter	BOOLEAN	Das Bogenfeld wird in Regeln als TDS-Listefeld (Abschnitt 2.3.6) verwendet (Voranstellen des @-Zeichens vor Feldnamen).

### Syntax der Feldgruppenregeln

In den Tabellen `FeldGruppe` bzw. `FeldgruppeFelder` sind die positiven Bedingungen für das Filterbogenfeld bzw. die abhängigen Bogenfelder einer Feldgruppe definiert. Jede Bedingung hat folgenden Aufbau:

`<Operator> <Operand>`

Der linke Operand wird hier weggelassen, weil er immer der Name des jeweiligen Bogenfeldes ist. Die komplette Bedingung für das Bogenfeld einer Feldgruppe lautet also:

`<Bogenfeld> <Operator> <Operand>`

Als Operator kann jeder dyadische Operator der Tabelle 34 verwendet werden. Die auf der rechten Seite erlaubten Operanden sind nachfolgend aufgelistet:

- Literale (Tabelle 33)
- LEER  
Kodelisten, in denen auch die Codes eines Schlüssels referenziert werden können;  
Beispiel: ( 1 ; 2 ; 3 )
- ICD-Listen oder OPS-Listen



#### Hinweis

Der rechte Operand darf kein Bogenfeld sein, da sich eine Feldbedingung immer genau auf ein Bogenfeld bezieht.

### Formale Definition von Feldgruppen

A sei ein Bogenfeld in einer Feldgruppe. Dann seien  $p(A)$  die positiven und  $n(A)$  die negativen Bedingungen, welche jeweils das Ergebnis wahr oder falsch haben können.

Eine Feldgruppe kann ggf. ein Filterfeld haben, das mit  $F$  bezeichnet wird. Eine Feldgruppe lässt sich dann in folgender Tabelle darstellen:

Tabelle 38: Formale Definition einer Feldgruppe

Feld	Positive Bedingung	Negative Bedingung	Bemerkung
F	$p(F)$	$n(F)$	falls Feldgruppentyp mit Filter
A1	$p(A1)$	$n(A1)$	
A2	$p(A2)$	$n(A2)$	
A3	$p(A3)$	$n(A3)$	
...			
An	$p(An)$	$n(An)$	

Eine Feldgruppe besteht insgesamt aus  $n$  abhängigen Bogenfeldern:

$A1, A2, \dots, An$

In Abhängigkeit von den Feldgruppentypen werden unterschiedliche Einzelregeln generiert.

#### Feldgruppen mit Filter

- Regeln der Feldgruppe „Optionale Mehrfachauswahl mit Filterfeld“ (MF\_OPTIONAL\_FILTER)

$$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$$

Insgesamt sind  $n$  Einzelregeln mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Obligatorische Mehrfachauswahl mit Filterfeld“ (MF\_MINDESTENS1\_FILTER)

$$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$$

$$p(F) \text{ UND } n(A1) \text{ UND } n(A2) \text{ UND } \dots \text{ UND } n(An)$$

Insgesamt sind  $n+1$  Einzelregeln mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Mehrfachauswahl mit Filterfeld, alle abhängigen Felder müssen positiv beantwortet sein“ (MF\_ALLES\_FILTER)

$$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$$

$$p(F) \text{ UND } n(A_i) \quad i = 1, \dots, n$$

Insgesamt sind  $2n$  Einzelregeln mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Einfachauswahl mit Filter“ (EF\_FILTER)

$$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$$

$$p(F) \text{ UND } n(A1) \text{ UND } n(A2) \text{ UND } \dots \text{ UND } n(An)$$

$$p(F) \text{ UND } p(A_j) \text{ UND } p(A_i) \quad \text{für alle unterschiedlichen } i, j = 1, \dots, n$$

Insgesamt sind  $n(n+1)/2+1$  Einzelregeln mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Optionale Einfachauswahl mit Filter“ (EF\_OPTIONAL\_FILTER)

$$n(F) \text{ UND } p(A_i) \quad i=1,\dots, n$$

$$p(F) \text{ UND } p(A_j) \text{ UND } p(A_i) \quad \text{für alle unterschiedlichen } i,j = 1,\dots, n$$

Insgesamt sind  $n(n+1)/2$  Einzelregeln mit der Feldgruppe verknüpft.

### Feldgruppen mit Filter: Attribut `nurPositiv`

Wenn in einer Feldgruppe mit Filter das Attribut `nurPositiv` gesetzt ist, so sind nur die Einzelregeln mit positiver Filterbedingung Bestandteil der Feldgruppe.

#### Beispiel:

Die Feldgruppe `EF_FILTER` mit `nurPositiv=ja` hat folgende Einzelregeln:

$$p(F) \text{ UND } n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$$

$$p(F) \text{ UND } p(A_j) \text{ UND } p(A_i) \quad \text{für alle unterschiedlichen } i,j = 1,\dots, n$$

### Feldgruppen ohne Filter

- Regeln der Feldgruppe „Einfachauswahl“ (`EF`)

$$n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$$

$$p(A_j) \text{ UND } p(A_i) \quad \text{für alle unterschiedlichen } i,j = 1,\dots, n$$

Insgesamt sind  $n(n-1)/2+1$  Einzelregeln mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Obligatorische Mehrfachauswahl“ (`MF_MINDESTENS1`)

$$n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$$

Insgesamt ist eine Einzelregel mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Und-Regel“ (`UND`)

$$p(A_1) \text{ UND } p(A_2) \text{ UND } \dots \text{ UND } p(A_n)$$

Insgesamt ist eine Einzelregel mit der Feldgruppe verknüpft.

### Feldgruppen mit mehreren Filterfeldern

Es besteht die Möglichkeit, Feldgruppen mit mehr als einem Filterfeld zu definieren:

Formal gibt es dann die Filterfelder  $F_1, F_2, \dots, F_n$  mit den positiven bzw. negativen Bedingungen  $p(F_j)$  bzw.  $n(F_j)$ . Für alle Filterfelder wird eine positive Bedingung  $p(F_1, \dots, F_n)$  und eine negative Bedingung  $n(F_1, \dots, F_n)$  gebildet. Diese modifizierten Filterbedingungen ersetzen die im Abschnitt B.2.3.7 definierten Filterbedingungen  $p(F)$  und  $n(F)$  bei den Einzelregeln.

Die Filterfelder können entweder über eine ODER-Verknüpfung oder eine UND-Verknüpfung miteinander verbunden sein:

$$p(F_1, \dots, F_n) = p(F_1) \text{ ODER } p(F_2) \text{ ODER } \dots \text{ ODER } p(F_n)$$

(ODER-Verknüpfung)

$$p(F_1, \dots, F_n) = p(F_1) \text{ UND } p(F_2) \text{ UND } \dots \text{ UND } p(F_n)$$

(UND-Verknüpfung)

Der Verknüpfungstyp ist im Attribut `fkFilterFeldTyp` der Tabelle `FeldGruppe` hinterlegt.

### Layout-Feldgruppen

Feldgruppen, bei denen das Attribut `grauWennNegativ` in der Datenbanktabelle `FeldGruppe` WAHR ist, werden nachfolgend als Layout-Feldgruppen bezeichnet. Der Attributname `grauWennNegativ` wurde gewählt, weil die abhängigen Felder der Layout-Feldgruppen auf den generierten Dokumentationsbögen eingegraut sind.

Layout-Feldgruppen haben folgende Eigenschaften:

- Sie haben mindestens ein Filterfeld.
- Jedes abhängige Feld hat die Bedingung `<> LEER` oder `EINSNICHTIN (LEER)` (Attribut `bedingung` in Tabelle `FeldGruppeFelder`).
- Das Attribut `nurPositiv` hat den Wert `FALSCH`.

Bei Vorliegen dieser drei Eigenschaften müssen die abhängigen Felder leer bleiben, wenn die negative Filterbedingung bei der Dokumentation eines Falles erfüllt ist.

### 2.3.8 Prüfung von Feldeigenschaften

Die in diesem Abschnitt behandelten feldbezogenen Prüfungen ergeben sich direkt aus den Tabellen `Feld` (bzw. `ErsatzFeld` oder `ZusatzFeld`) und `BogenFeld` (bzw. `ExportFormat`) und werden vor Evaluation der in Abschnitt B 2.3.4 beschriebenen feldübergreifenden Regeln durchgeführt.

Die hier beschriebenen Prüfungen sind nur in Form von Feldeigenschaften – nicht aber in Regelsyntax – in der Datenbank für QS-Dokumentation hinterlegt.

### Überprüfung des Formats

Die Formatprüfung bezieht sich auf den Exportdatensatz (Abschnitt B 2.4.3): Die QS-Dokumentations- bzw. Exportsoftware muss Daten im korrekten Format generieren, was durch die datenentgegennehmenden Stellen geprüft wird. Für Exportfelder (Tabelle `ExportFormat`), die einen Bezug zu einem Ersatzfeld (Tabelle `ErsatzFeld`) bzw. zu einem Zusatzfeld (Tabelle `ZusatzFeld`) haben, sind die standardisierten Fehlertexte anzupassen.

Die Prüfung bezieht sich insbesondere darauf, ob der Feldinhalt dem in der Spezifikation definierten Basistyp entspricht. Beispielsweise sind Buchstaben beim Basistyp `GANZEZAHL` nicht erlaubt.

### Standardisierter Fehlertext für Formatfehler eines Ersatzfeldes

Der Wert '`<WERT>`' des Datenfeldes `<Modul.name>:<Bogen.name>:<Ersatzfeld.name>` '`<ErsatzFeld.bezeichnung>`' ist kein gültiger `<BasisTyp.name>` Wert (`<BasisTyp.bezeichnung>` `<BasisTyp.format>`).

**Standardisierter Fehlertext für Formatfehler eines Bogenfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<Feld.name> '<BogenFeld.bezeichnung>' (Zeile <BogenFeld.zeileAufBogen>) ist kein gültiger <BasisTyp.name> Wert (<BasisTyp.bezeichnung> <BasisTyp.format>).

**Standardisierter Fehlertext für Muss-Fehler eines Zusatzfeldes**

Das Zusatzfeld <Modul.name>:<Bogen.name>:<ZusatzFeld.name> '<ZusatzFeld.bezeichnung>' ist kein gültiger <BasisTyp.name> Wert (<BasisTyp.bezeichnung> <BasisTyp.format>).

**Überprüfung der Feldlänge**

Die Feldlängenprüfung bezieht sich darauf, ob die Anzahl der Zeichen eines Wertes die spezifizierte Länge<sup>51</sup> (Attribut laenge in Tabelle Feld) des Feldes überschreitet.

**Standardisierter Fehlertext für Längenfehler eines Ersatzfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' überschreitet die zulässige Feldlänge <Feld.laenge>.

**Standardisierter Fehlertext für Längenfehler eines Bogenfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<Feld.name> '<BogenFeld.bezeichnung>' (Zeile <BogenFeld.zeileAufBogen>) überschreitet die zulässige Feldlänge <Feld.laenge>.

**Standardisierter Fehlertext für Muss-Fehler eines Zusatzfeldes**

Das Zusatzfeld <Modul.name>:<Bogen.name>:<ZusatzFeld.name> '<ZusatzFeld.bezeichnung>' überschreitet die zulässige Feldlänge <ZusatzFeld.laenge>.

**Überprüfung der Schlüsselcodes**

Die Überprüfung von Schlüsselcodes bezieht sich darauf, ob bei Schlüsselfeldern nur zulässige Schlüsselcodes verwendet werden.

**Standardisierter Fehlertext bei unzulässigen Schlüsselcodes eines Ersatzfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' ist kein zulässiger Code des Schlüssels <Schlüssel.name> (<Schlüssel.bezeichnung>).

**Standardisierter Fehlertext bei unzulässigen Schlüsselcodes eines Bogenfeldes**

---

<sup>51</sup> Wenn bei einem Ersatz die Länge nicht spezifiziert ist, so entfällt die Prüfung.

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<Feld.name> '<BogenFeld.bezeichnung>' (Zeile <BogenFeld.zeileAufBogen>) ist kein zulässiger Code des Schlüssels <Schluessel.name> (<Schluessel.bezeichnung>).

#### **Standardisierter Fehlertext für Muss-Fehler eines Zusatzfeldes**

Das Zusatzfeld <Modul.name>:<Bogen.name>:<ZusatzFeld.name> '<ZusatzFeld.bezeichnung>' ist kein zulässiger Code des Schlüssels <Schluessel.name> (<Schluessel.bezeichnung>).

#### **Überprüfung numerischer Wertebereiche**

Bei numerischen Datenfeldern (Typ ZAHL oder GANZEZAHL) ist hart zu überprüfen, ob der Zahlenwert im durch die Attribute min und max (Tabelle FELD) definierten Wertebereich liegt:

- wert < min (nur prüfen, wenn min <> LEER)
- wert > max (nur prüfen, wenn max <> LEER)

In der Tabelle Feld sind weitere Unter- und Obergrenzen (Attribute min/max und minWeich/maxWeich) für Prüfungen definiert. In wenigen Fällen sind auch in der Tabelle BogenFeld Grenzen definiert. Falls vorhanden (= not null), werden die in der Tabelle BogenFeld gesetzten speziellen Wertgrenzen statt der Grenzen in der Tabelle Feld angewandt.

Eine Übersicht über die in numerischen Datenfeldern definierten harten und weichen Wertebereiche bietet die Abfrage WertebereicheNumerischerFelder.

Außerdem sind die Wertebereiche in den Ausfüllhinweisen zu den jeweiligen Datenfeldern angegeben.

#### **Standardisierter Fehlertext bei Unterschreitung einer Wertebereichsgrenze eines Ersatzfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' ist kleiner als '<Feld.min>'.

#### **Standardisierter Fehlertext bei Unterschreitung einer Wertebereichsgrenze eines Bogenfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<Feld.name> '<BogenFeld.bezeichnung>' (Zeile <BogenFeld.zeileAufBogen>) ist kleiner als '<Feld.min>'.

#### **Standardisierter Fehlertext bei Überschreitung einer Wertebereichsgrenze eines Ersatzfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' ist größer als '<Feld.max>'.

#### **Standardisierter Fehlertext bei Überschreitung einer Wertebereichsgrenze eines Bogenfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<Feld.name> '<BogenFeld.bezeichnung>' (Zeile <BogenFeld.zeileAufBogen>) ist größer als '<Feld.max>'.

Bei weichen Plausibilitätsverletzungen ist dem Fehlertext das Wort „Hinweis“ voranzustellen.

## Überprüfung der Muss-Felder

Ein nicht ausgefülltes Muss-Feld (Abschnitt B 2.2.3) führt zu einer Regelverletzung.

### Standardisierter Fehlertext für Muss-Fehler eines Ersatzfeldes

Das Datenfeld '`<Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>'` muss einen gültigen Wert enthalten.

### Standardisierter Fehlertext für Muss-Fehler eines Bogenfeldes

Das Datenfeld '`<Modul.name>:<Bogen.name>:<Feld.name> '<BogenFeld.bezeichnung>'` (Zeile `<BogenFeld.zeileAufBogen>`) muss einen gültigen Wert enthalten.

### Standardisierter Fehlertext für Muss-Fehler eines Zusatzfeldes

Das Zusatzfeld `<Modul.name>:<Bogen.name>:<ZusatzFeld.name> '<ZusatzFeld.bezeichnung>' muss einen gültigen Wert enthalten.`

## 2.3.9 Verfahren für die Evaluation von Regeln

Grundsätzlich muss jede gem. Abschnitt B 2.3.2 formulierte Regel evaluiert werden, wenn keine der folgenden drei Bedingungen zutrifft:

1. Für mindestens ein referenziertes Bogenfeld<sup>52</sup> schlägt eine harte Feldprüfung (Abschnitt B 2.3) fehl.<sup>53</sup>
2. Ein Feld der Regel ist nicht ausgefüllt (LEER) und **keine** der folgenden Teilbedingung trifft in Bezug auf das leere Feld zu:
  - Es ist in einer Liste enthalten, die mit einem Listenoperator (EINSIN, KEINSIN, JEDESIN, EINSNICHTIN) geprüft wird bzw. wird direkt gegen eine Liste geprüft (IN, NICHTIN).
  - Es wird in der Regel explizit mit `<>` oder `=` auf LEER geprüft.
  - Das Feld der Regel befindet sich auf einem vorhandenen Teildatensatz.<sup>54</sup>
  - Jeder Operation auf einen Wert `<> LEER` ist eine ODER-Verknüpfte Prüfung auf LEER direkt vorgeschaltet (`Feld = LEER ODER Feld Operator Operand`).
3. Eine Funktion der Regel hat das Ergebnis LEER und wird in der Regel nicht explizit mit `<>` oder `=` auf LEER geprüft.

Der Algorithmus zur Evaluation einer Plausibilitätsregel ist in Abbildung 4 beschrieben.

<sup>52</sup> Die Verbindung zwischen Regeln und Bogenfeldern geschieht über die Tabelle `RegelFelder`, siehe Regelfelder (Bogenfelder einer Regel), Abschnitt B 2.3.1.

<sup>53</sup> Erst bei Fehlerfreiheit der feldbezogenen Prüfungen werden die feldübergreifenden Prüfungen durchgeführt.

<sup>54</sup> Eine Regel wird nicht geprüft, wenn der entsprechende Teildatensatz eines leeren Feldes optional und im konkreten Fall nicht vorhanden ist.

### Umgang mit Laufzeitfehlern

Bei der Evaluation von Regeln können, z.B. bei einem Vergleich von LEER < WERT, Laufzeitfehler entstehen. Solche Laufzeitfehler sind bei der Evaluation zu verhindern.

Laufzeitfehler bei der Evaluation von Regeln nach den Vorgaben sind immer ein Hinweis auf Syntaxfehler in der Regel. Das Ausbleiben von Laufzeitfehlern ist noch kein Beweis für die syntaktische Fehlerfreiheit einer Regel, da es auch fehlertolerante Parser geben könnte, die beim Verlassen des definierten Wertebereichs der Regel ein FALSCH zurückgeben können. Tatsächlich liegt aber hier ein undefinierter Zustand vor, der weder FALSCH noch WAHR ist.

Ein undefinierter Zustand an einem beliebigen Punkt im Evaluationsprozess muss für alle dort noch möglichen Werte durch die Struktur der Regel immer vermieden werden.

Algorithmus zur Evaluation einer Plausibilitätsregel

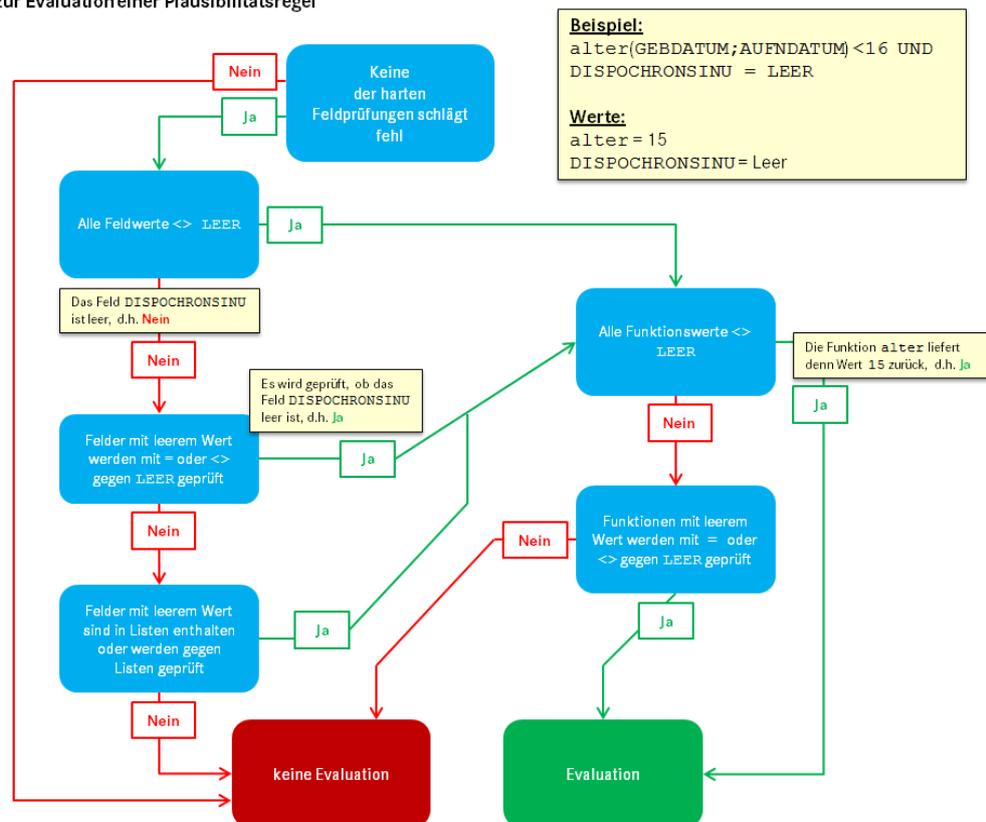


Abbildung 4: Algorithmus zur Evaluation von Plausibilitätsregeln

### Teildatensatzübergreifende Regeln

Teildatensatzübergreifende Regeln (Abschnitt B 2.3.6) müssen u.U. mehrfach evaluiert werden (für jede Kombination von Teildatensätzen, die von der Regel betroffen ist).



**Hinweis**

In wenigen Einzelfällen beziehen sich Plausibilitätsregeln auf mehr als zwei Teildatensätze.

## 2.4 Exportfeldbeschreibung

Neben der Datenfeldbeschreibung (Abschnitt B 2.2) enthält die Spezifikationsdatenbank die Beschreibung der Exportfelder für ein Modul (Exportdatensatz). Diese werden zum Teil über Ersatzfelder berechnet. Es wird zudem zwischen Dokumentationsmodulen (Abschnitt B 2.2.1) und Exportmodulen unterschieden.

### 2.4.1 Exportmodule

Um die unterschiedlichen Abrechnungswege und davon abhängige Datenflüsse mit unterschiedlichen Datenannahmestellen für die Datenübermittlung zu berücksichtigen, wird zwischen den Dokumentationsmodulen (Tabelle `Modul`) und den Exportmodulen (Tabelle `ExportModul`) unterschieden.

Tabelle 39: Struktur der Tabelle `ExportModul`

Feldname	Datentyp	Bemerkung
<code>idExportModul</code>	INTEGER	Primärschlüssel
<code>fkModul</code>	INTEGER	Bezug zum Dokumentationsmodul
<code>name</code>	TEXT	Technischer Name (muss eindeutig sein)
<code>bezeichnung</code>	TEXT	Bezeichnung des Exportmoduls
<code>existenzBedingung</code>	TEXT	Definiert, unter welcher Bedingung das Modul in ein definiertes Exportmodul transformiert wird.
<code>type_QS_data</code>	TEXT	Datentyp im XML-Schema
<code>ersatzBedingungMDS</code>	TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
<code>type_QS_data_mds</code>	TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.

Das Attribut `type_QS_data` gibt Auskunft darüber, welchem Datentyp dies im XML-Schema entspricht.

#### Beispiel:

Beim Exportmodul `NWIEA_LKG` ist der Datentyp im XML-Schema `qs_data_nwiea_type`.

Softwareanbietern soll hiermit die Integration eines Mechanismus der automatischen Datentypzuweisung ermöglicht werden, um den Aufwand zu reduzieren und Fehler zu vermeiden.

## 2.4.2 Modulauslöser, Dokumentations- und Exportmodule

### Modulauslöser

In der Spezifikationsdatenbank für den QS-Filter (QSF) erfolgt z.B. für das Dokumentationsmodul NWIEA eine Aufteilung der Modulauslöser analog zu den zwei verschiedenen Exportmodulen NWIEA\_LKG und NWIES\_LKG (Abschnitt B 2.4.1).

- Der Modulauslöser NWIES\_LKG löst den QS-Datensatz NWIES aus.
- Der Modulauslöser NWIEA\_LKG löst den QS-Datensatz NWIEA aus.

Hier wird ein konkreter Bezug zur Abrechnungsart hergestellt. Für NWIEA\_LKG und NWIES\_LKG werden hierbei alle Fälle gemäß § 301 berücksichtigt, die das administrative Einschlusskriterium erfüllen.

### Dokumentationsmodule

In der Spezifikationsdatenbank für die QS-Dokumentation sind die Dokumentationsmodule NWIEA und NWIES definiert. Sowohl für den Krankensektor als auch für den vertragsärztlichen Sektor werden jeweils die gleichen Dokumentationsbögen ausgefüllt.

- Das Dokumentationsmodul NWIES wird von Einrichtungen, die stationäre Fälle am Krankenhaus behandeln oder stationäre belegärztliche Leistungen erbringen, dokumentiert.
- Das Dokumentationsmodul NWIEA wird von Einrichtungen, die ambulante Fälle am Krankenhaus behandeln oder ambulante vertragsärztliche Leistungen erbringen, dokumentiert.

Über die folgenden Datenfelder im Dokumentationsbogen gibt der Dokumentierende an, ob sich der Datensatz auf den vertragsärztlichen Sektor oder auf den Krankensektor bezieht.

Tabelle 40: Datenfelder zum Status des Leistungserbringers

idBogenFeld	Bogenfeldbezeichnung	Feld	Schlüssel
8601	Status des Leistungserbringers	NWIEA.B.1. STATUSLEA	1 = Krankenhaus 2 = Vertragsarzt
8568	Status des Leistungserbringers	NWIES.B.1. STATUSLES	1 = Krankenhaus 2 = Belegarzt nach § 121

**Hinweis****Vorbelegung der Datenfelder zum Status des Leistungserbringers**

Bei den durch die Kassenärztlichen Vereinigungen zur Verfügung gestellten (z. B. webbasierten) Anwendungen zur Erfassung und Übertragung von Daten der einrichtungsbezogenen QS-Dokumentation ist immer der Schlüsselwert 2 anzugeben. Dies ist der Fall, wenn die Dokumentationspflicht durch die Modulauslöser `NWIEA_KV` oder `NWIES_KV` durch die jeweilige Kassenärztliche Vereinigung identifiziert wurde.

Für den Krankenhaussektor (im Abrechnungskontext) ist immer der Schlüsselwert 1 anzugeben. Dies ist der Fall, wenn die Dokumentationspflicht durch die Modulauslöser `NWIEA_LKG` oder `NWIES_LKG` identifiziert wurde.

Die Datenfelder dürfen in der jeweiligen Softwarelösung daher entsprechend vorbelegt werden.

---

**Exportmodule**

Nach Abschluss der Dokumentation wird das Dokumentationsmodul je nach Eintrag im Datenfeld „Status des Leistungserbringers“ in das entsprechende Exportmodul transformiert.

---

**Beispiel:**

Das Modul `NWIES` gliedert sich nach Abschluss der Dokumentation bei Erfüllen der Existenzbedingung in zwei Exportmodule, die jeweils einem separaten Datenfluss entsprechen (`NWIES_LKG`, `NWIES_KV`). Wird im Dokumentationsmodul `NWIES` im Feld `STATUSLES` beispielsweise 1 = Krankenhaus kodiert, wird dieses nach Abschluss der Dokumentation in das Exportmodul `NWIES_LKG` transformiert. Damit wird die korrekte Datenannahmestelle definiert, an die der Datensatz zu übermitteln ist – in diesem Fall die `LQS/LKG`.

---

**2.4.3 Exportdatensatz**

Der Exportdatensatz enthält die Exportfelder für ein Modul. Welche Zusatzfelder, Bogenfelder und/oder Ersatzfelder den Exportdatensatz pro Modul bilden, ist in Tabelle `ExportFormat` definiert. Weiterführende Informationen zum Exportdatensatz sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

**2.5 Versionierung**

Im Folgenden werden die Tabelle `Version`, der Abgleich zu vorherigen Versionen, die Abgrenzung zwischen Erfassungsjahren und Datensatzformaten sowie die Version von Exportverfahren und -dateien beschrieben.

**2.5.1 Grundlegende Definitionen**

In der Tabelle `Version` finden sich Informationen zur Version der Spezifikationsdatenbank.

Die wichtigsten Eigenschaften einer Version sind der Versionsname (Attribut `name`) und der Gültigkeitszeitraum (Attribute `ab` und `bis`).

**Achtung**

Der Gültigkeitszeitraum einer Version ist in der fallbezogenen QS-Basispezifikation für Leistungserbringer in der Regel ein Erfassungsjahr (z.B. Aufnahme zwischen dem 01.01.2017 und dem 31.12.2017). Die Auslösung der einrichtungsbezogenen QS-Dokumentation bezieht sich bei den Modulen `NWIEA` und `NWIES` nur auf **die ersten beiden Quartale des Erfassungsjahres**.

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation ist beispielsweise für das Erfassungsjahr 2017 vom 01.01.2017 bis zum 30.06.2017 gültig. Die Prozesse des Verfahrens können außerhalb dieses Zeitraums liegen.

---

Die QS-Dokumentationssoftware eines Erfassungsjahres wird für diejenigen Einrichtungen verwendet, die Fälle mit Aufnahmedatum in das Krankenhaus (stationäre Fälle) bzw. mit Behandlungsdatum (ambulante Fälle) im oben definierten Gültigkeitszeitraum behandeln.

Jedes Modul der Datenbank hat eine Version (vgl. Attribut `fkVersion` in Tabelle `Modul`). In einer Spezifikationsdatenbank können Module unterschiedlicher Versionen enthalten sein. Über die in der Datenbank definierten Relationen sind auch für alle Bogenfelder (Tabelle `Bogenfeld`), Exportfelder (Tabelle `ExportFormat`) und Plausibilitätsregeln (Tabelle `Regeln`) Versionen definiert.

**Status der Spezifikation**

Versionen können den Status `in Entwicklung`, `final` oder `Update der finalen Spezifikation` haben. Diese Zustände werden zum Nachschlagen in der Tabelle `VersStatus` verwaltet. Das Attribut `gueltig` zeigt die gültige Version der Datenbank an. Es darf nur eine einzige Version als gültig markiert sein.

---

**Beispiel:**

Die Version 2017 V01, einschließlich ihrer Updates, sind finale Versionen.

---

Hat eine Spezifikationsdatenbank den Status `in Entwicklung`, kann `Modul.fkVersion` als `ungueltig` markierte Versionen enthalten, um Zwischenstände abzubilden.

**Historie der Versionen**

Die Tabelle `Version` enthält auch einen Selbstbezug (Attribut `fkVersion`), der die Identifizierung der Vorgängerversion ermöglicht.

**2.5.2 Delta-Informationen zur vorhergehenden Version**

Um den Benutzern der Spezifikation umfassende Informationen zu den jeweiligen Änderungen zur Verfügung zu stellen, enthält die Spezifikationsdatenbank Tabellen, die den Änderungsstand im Vergleich mit der letzten gültigen Version des Vorjahres und zur Vorversion der Datenbank

anzeigen. Diese sogenannten Delta-Tabellen werden automatisch generiert. Weiterführende Erläuterungen zu den Delta-Informationen zur vorhergehenden Version sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

## 2.6 Administrative Objekte

Die Datenbank für QS-Dokumentation beinhaltet neben den Dokumentationsobjekten eine Gruppe von administrativen Objekten, die Hilfestellungen für die Einhaltung eines korrekten Datenflusses und standardisierter Prüfprozesse geben.

Thematisch sind diese Objekte in drei Bereiche aufgeteilt:

1. Mapping-Informationen, mit denen QS-Daten, die außerhalb des eigentlichen QS-Containers (dem Element `<qs-daten>`) im XML verortet werden können.
2. Eine Übersicht über die für ein Modul innerhalb einer Region relevanten technischen Datenservices mit Empfängeradressen und zu verwendenden XML-Schlüsseln.
3. Eine Auflistung und Kategorisierung von Prüfschritten zur Implementierung eines Datenservices.

### 2.6.1 CSV/XML-Mapping in der Spezifikationsdatenbank (QSDOK)

Es gibt eine Reihe von XML-Elementen, die zurzeit nicht in der Access-DB hinterlegt sind. Insgesamt gibt es vier Kategorien von Abweichungen:

- Neu eingeführte XML-Elemente
- Header-Informationen, wie die GUID des Dokuments
- Umbenannte und aus dem QS-Datensatz ausgelagerte CSV-Felder
- Gelöschte Felder wie beispielsweise das Feld `IKNRKH`

Die CSV-Felder werden nicht an die XML-Struktur angepasst. Stattdessen wird für betroffene Felder eine Mapping-Tabelle in der Access-DB hinterlegt.

Die durch Bögen (Teildatensätze) für das Element `<qs_data>` festgelegte Struktur bleibt durch die aufgeführten Mappings unangetastet. Die Mappings erlauben alleine eine Festlegung darüber, ob und falls ja wo ein betroffenes Datum außerhalb des `<qs_data>` Elements in das XML geschrieben wird und ob das Datum für das Element `<qs_data>` gelöscht werden muss. Ggf. nötige Umbenennungen außerhalb von `<qs_data>` werden implizit durch die Ortsangabe vorgenommen.

Die für das Mapping geführten Tabellen heißen `ExportZiele` und `ExportZielXML`. In der Abfrage `vExportZieleXML` sind diese Tabellen zu einer Übersicht zusammengefasst. Hierbei ist zu beachten, dass nicht jedem Eintrag in der Tabelle `ExportZiele` auch ein Eintrag in der Tabelle `ExportZielXML` zugewiesen sein muss. Soll beispielsweise ein Datum aus dem Element `<qs_data>` lediglich gelöscht werden, ohne jedoch an andere Stelle im XML verschoben zu werden, ist keine dementsprechende `xmlXPath`-Angabe erforderlich.

Es können alle Feldarten referenziert sein. Referenziert sind aber nur solche Felder, die nicht oder nicht nur im Element `<qs_data>` aufgeführt werden. Die Felder werden referenziert und über diese Tabelle mit zusätzlichen Informationen in Bezug auf das XML verknüpft.

Die über das Attribut `fkFeld` referenzierten Felder gelten für alle Module, in denen dieses Feld verwendet wird. Wird ein Feld (gleichzeitig) über unterschiedliche Feldarten referenziert, überlagern detailliertere Angaben die allgemeinen. Wird so beispielsweise in der Tabelle `ExportZiele` ein Feld allgemein über den Fremdschlüssel `fkFeld` und in einer weiteren Zeile modulspezifisch über `fkBogenFeld` referenziert, werden die allgemeinen Regelungen (`loeschenQS`, `exportWennLeer`, `xmlXPath` und `xmlGruppierung`) für das Feld mittels der Referenzierung über `fkFeld` mit den spezifischeren Regelungen für das modulbezogene Feld mittels der Referenzierung über `fkBogenFeld` für das konkret aus der Referenzierung hervorgehende Modul überschrieben.

Dabei ist ebenfalls eine etwaige Konkretisierung über `fkExportModul` zu beachten, welche Ihrerseits konkretisierenden Charakter hat. Die über das Attribut `fkErsatzFeld` referenzierten Ersatzfelder gelten für alle Exportmodule, in denen dieses Ersatzfeld verwendet wird. Bei Referenzierung des gleichen Feldes (insbesondere bezogen auf `fkFeld`, `fkErsatzFeld` bzw. `fkZusatzfeld`) in zwei Zeilen - einmal ohne und einmal mit Angabe von `fkExportmodul` - so überschreibt der Eintrag mit Angabe den Eintrag ohne Angabe für das konkret benannte Exportmodul.

Die booleschen Spalten `loeschenQS` und `exportWennLeer` geben folgende Informationen an:

- `loeschenQS`: Das Datum wird nur an den/die alternativen Ort(e) geschrieben und taucht in den QS-Daten nicht mehr auf. Hier geht es zum Beispiel um vom Datenschutz betroffene Felder, die nur in bestimmte Bereiche des XML geschrieben werden dürfen, wo sie dann verschlüsselt werden können.
- `exportWennLeer`: Hier wird ein leeres Feld nur dann berücksichtigt und leer eingetragen, wenn dieser Wert auf `true` steht.

In der Spalte `xmlXPath` vom Typ String steht ein XPath-Ausdruck, der ein Mapping auf einen im XML liegenden Ort des betroffenen Feldes enthält. Ausgangspunkt des XPath-Ausdrucks ist immer der aktuelle Datensatz im XML, also das `<case>` Element.

Die boolesche Spalte `xmlGruppierung` gibt an, dass ein neuer Knoten nur eingesetzt wird, wenn das Datum einen neuen Wert enthalten sollte.

## 2.6.2 Datenservices

In den administrativen Objekten in der QSDOK-Datenbank sind Angaben über die beim Export relevanten technischen Datenservices und ihre verfahrensbezogene und regionale Zuordnung zu finden.

Die einzelnen Datenservices werden zusammen mit den E-Mail-Adressen<sup>55</sup>, an die die Exportdateien zu versenden sind, und den für die Verschlüsselung der QS-Daten zu verwendenden XML-Schlüsseln in der Tabelle `Datenservice` abgebildet.

---

<sup>55</sup> Die KVen und die DAS-SV erhalten die Daten in der Regel auf anderen Übertragungswegen.

Die einzelnen Datenannahmestellen, die ihre Datenannahme mithilfe eines oder mehrerer dieser Datenservices realisieren, sind in der Tabelle `Region` hinterlegt, während die Zuordnung eines Datenservices zu einer Region abhängig vom Modul in der Tabelle `DatenserviceModul` festgelegt ist.

**Hinweis****Administrative Objekte in der QSDOK-Datenbank**

Eine Auflistung und Kategorisierung von Prüfschritten zur Implementierung eines Datenservices ist in der vorliegenden Spezifikation aktuell nicht enthalten.

---

## 3 XML-Schema

Die XML-Schema-Datei (XSD) ist eine Empfehlung des W3C<sup>56</sup> zum Definieren von Strukturen für XML-Dokumente.

In der Spezifikation der einrichtungsbezogenen QS-Dokumentation werden die XML-Schemata aus der Spezifikationsdatenbank abgeleitet und haben die Aufgabe, die aktuellen Datenflussmodelle der G-BA-Richtlinien abzubilden:

- Abbildung der zu exportierenden QS-Daten
- Abbildung der Schnittstellen an den Institutionsübergängen (LE und DAS, DAS und BAS)
- Abbildung der vorzunehmenden Datentransformation in den jeweiligen Einrichtungen des Datenflusses  
(z.B. LE-Pseudonymisierung bei der DAS)
- Abbildung der Rückprotokollierung

Aus diesem Grund gibt es bei der Erstellung von Schemata, welche die Konformität von Richtlinie und Datenschutz sicherstellen sollen, kein „Allround-Schema“, welches alle Anforderungen an alle Beteiligten abdeckt, sondern eine „Schema-Familie“, aus der heraus gezielt für jede Schnittstelle („Interface“) eine passende Datenstruktur definiert wird.



### Hinweis

Im Folgenden erläutert sind die Abschnitte

- 3.1 „Kompositionsmodell“,
- 3.2 „Schnittstellen“ und
- 3.3 „Darstellung der XML-Struktur“.

Erläuterungen zum „Aufbau der XML-Exportdatei“ sind in der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer beschrieben.

---

### 3.1 Kompositionsmodell

Bei der Schema-Erstellung wurde auf ein Kompositionsmodell zurückgegriffen, in dem sich alle Teilschemata am Ende einen Namensraum teilen, um diese Schema-Familie besser warten zu können und gleiche Teilstrukturen nur einmal definieren zu müssen. Als Bezeichnung des Namensraums wurde „urn:gba:sgg“ gewählt. Zu diesem Namensraum werden die Bausteine je nach Bedarf über „includes“ zusammengestellt.

---

<sup>56</sup> <http://www.w3.org/XML/Schema>

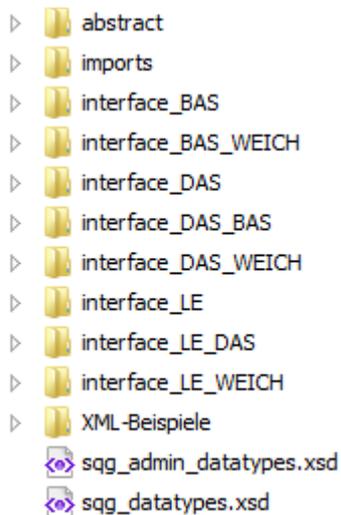


Abbildung 5: Dateiodner der Schnittstellen-Schemata

Das Kompositionsmodell ermöglicht es, Konzepte aus der objektorientierten Programmierung – darunter fallen die Konzepte abstrakter Typ, Ersetzbarkeit von Typen, Wiederverwendung und Polymorphismus – zu nutzen. Dadurch können Schemata erstellt oder genutzt werden, die generische Grundtypen definieren und diese Typen so erweitern, dass sie schnittstellenspezifisch sind, ohne das ursprüngliche Schema zu beeinflussen. Dieses Kompositionsmodell wird hier näher erläutert.

#### Beispiel:

Die leistungserbringeridentifizierenden Daten existieren gemäß Datenflussmodell der G-BA-RL in drei Ausprägungen:

- im Klartext (Schnittstelle LE)
- pseudonymisiert (Schnittstelle DAS)
- verschlüsselt (Schnittstelle DAS-BAS)

Bei dieser Konzeption werden alle drei Ausprägungen vom selben Basisdatentyp geerbt, in einem zweiten Schritt die drei Ausprägungen konkretisiert und angepasst, und über „includes“ in die jeweilige Schnittstelle integriert.

### 3.2 Schnittstellen

In der folgenden Tabelle werden Schema-Dateien aufgeführt, die im Rahmen der Übermittlung der QS-Daten Verwendung finden. Andere Dateien haben zwar ebenfalls die Dateiendung „.xsd“, sind aber keine vollständigen Schemata, sondern Bausteine für Schnittstellen.

Tabelle 41: Verwendbare Schemata und Ablageort

Schnittstelle	Schema	Verwendung für	Absender	Empfänger
LE interface_LE	interface_LE_kv	Exportdateien für Datenfluss über KV (ohne XML-Verschlüsselung)	KH/PR-Kollektiv	KV

Schnittstelle	Schema	Verwendung für	Absender	Empfänger
	interface_LE_Iqs	Exportdateien für Datenfluss über LQS/LKG (ohne XML-Verschlüsselung)	KH	LQS
LE/DAS interface_LE_DAS	interface_LE_KV	Transferdateien für Datenfluss über KV (Verschlüsselung mit BAS-Schlüssel von feedback_key und qs_data)	PR-Kollektiv	KV
	interface_LE_LQS	Transferdateien für Datenfluss über LQS (Verschlüsselung mit DAS-Schlüssel von qs_data)	KH	LQS
	response_DAS_LE	Datenflussprotokoll der DAS	DAS	LE
	response_receipt	Empfangsbestätigung der DAS	DAS	LE
DAS interface_DAS	interface_KV	Verarbeitungsdateien	Nach der LE-Pseudonymisierung	
	interface_LQS	Verarbeitungsdateien	Nach der QS-Entschlüsselung	
	interface_LQS_psn	Verarbeitungsdateien	Nach der QS-Entschlüsselung und LE-Pseudonymisierung	
DAS/BAS interface_DAS_BAS	interface_DAS_BAS	Transferdateien (Verschlüsselung mit BAS-Schlüssel von qs_data, care_provider und ggf. feedback_key)	DAS	BAS
	response_BAS_DAS	Datenflussprotokoll der BAS	BAS	DAS
BAS Interface_BAS	interface_BAS.xsd	Verarbeitungsdateien	Nach der Entschlüsselung	

Um nach einer Schemavalidierung der XML-Dateien die Weiterverarbeitung und dementsprechend die spezifikationskonforme Protokollierung auf Datensatzebene weiterhin zu ermöglichen, wurden neben der oben in Tabelle 41 beschriebenen Schemavariante ein „weiches“ Schema für die Schnittstellen LE und DAS eingeführt. Diese weiche Variante wird ausschließlich mit dem Datenprüfprogramm verwendet. Es ist dabei zu beachten, dass nur die DAS, die die QS-Daten entschlüsseln, diese weiche Variante benötigen. Diese sind LQS/LKG und BAS.

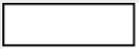
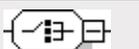
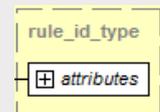
In der folgenden Tabelle werden die Schema-Dateien aufgeführt, die im Rahmen der Prüfung mit dem Datenprüfprogramm verwendet werden.

Tabelle 42: Weiche Schemavarianten für das Datenprüfprogramm

Schnittstelle	Schema	Verwendungszweck in Zusammenhang mit dem DPP
LE Interface_LE_WEICH	interface_LE_kv	Validierung der Module für den Datenfluss über die KV als DAS mit dem Datenprüfprogramm (DPP)
	interface_LE_lqs	Validierung der Module für den Datenfluss über die LQS mit dem Datenprüfprogramm (DPP)
DAS Interface_DAS_WEICH	interface_LQS	Validierung mit dem DPP (nach der Entschlüsselung der QS-Daten)
BAS interface_BAS_WEICH	interface_BAS	Validierung mit dem Datenprüfprogramm (DPP)

### 3.3 Darstellung der XML-Struktur

Zur Veranschaulichung der verwendeten XML-Schemata werden Diagramme verwendet, deren Symbole in der folgenden Tabelle kurz dargestellt und erläutert werden.

Symbol	Beschreibung
	<b>Optionales Element</b> Kardinalität 0..1 („0 oder 1“)
	<b>Obligatorisches Element</b> Kardinalität 1: das Element muss genau einmal vorkommen
	<b>Mehrfach wiederholbares Element</b> Kardinalität: die erlaubte Anzahl der Elemente wird unter dem Symbol dargestellt (Beispiel: 1..n, n..m).
	<b>Referenzelement</b> Das referenzierte globale Element ist an anderer Stelle im Schema definiert.
	<b>Eine Folge von Elementen</b> Die Elemente müssen genau in der Reihenfolge vorkommen, in der sie im Schemadiagramm angezeigt sind.
	<b>Eine Auswahl von Elementen</b> Nur ein einziges Element aus der Liste kann ausgewählt werden.
	Ein Element mit Kind-Elementen
	<b>Komplexer Datentyp</b> Der komplexe Datentyp wird mit einem Rahmen mit einem gelben Hintergrund angezeigt.

## 4 Tools

Das vorliegende Kapitel beschreibt Spezifikationskomponenten, die als Hilfsprogramme Prozesse in der Qualitätssicherung unterstützen. Die Hilfsprogramme basieren auf der Programmiersprache Java. Dementsprechend ist ein Abschnitt enthalten, der die Installation einer Java-Laufzeit-Umgebung (JRE) beschreibt.

Das **Verschlüsselungspaket** wird in Abschnitt 4.3 erläutert. Informationen zum **Datenprüfprogramm** sind in Abschnitt 4.2 beschrieben. Das **LE-Pseudonymisierungsprogramm** ist im in der Spezifikationskomponente enthaltenen Anwenderhandbuch und in der dazugehörigen Datenflussspezifikation beschrieben (Abschnitt 4.4).

### 4.1 Java-Installation

Zur Ausführung von Java-Programmen wird das Java Runtime Environment (JRE) zur Verfügung gestellt. Das JRE kann „online“ und „offline“ installiert werden.

Offizielle Installationspakete können von der Website <http://www.java.com/de/download/manual.jsp> heruntergeladen werden.

Anleitungen zur Installation von Java unter verschiedenen Betriebssystemen sind auf der folgenden Website detailliert beschrieben:

[https://www.java.com/de/download/help/download\\_options.xml](https://www.java.com/de/download/help/download_options.xml)

#### Aufruf

Der Aufruf des Java-Programms erfolgt über die Konsole des Betriebssystems. Die Rückmeldungen des Programms erfolgen ebenfalls über die Konsole und können dort abgefangen werden.

Die Ein- und Ausgabe der Konsole lässt sich in der Regel in jedes Programm integrieren. Ein Exitcode von 0 ohne Ausgabe entspricht einer fehlerfreien Verarbeitung des Programms. Bei Fehlern gibt es einen Exitcode von 1 und in der Regel eine Konsolen- bzw. eine Error-Ausgabe.

Aufruf: `java -jar Beispielprogramm.jar -g -o Dateiname 2> error.txt`

- Java: Aufruf der JVM mit dem Befehl `java`
- `-jar`: Parameter `-jar`, mit dem der JVM mitgeteilt wird, dass ein Java-Archiv aufgerufen wird
- `Beispielprogramm.jar`: Benennung des Archivs, mit vollem Pfad, wenn es nicht im aktuellen Verzeichnis liegt
- `-g -o`: Parameter, die an das Java Programm übergeben werden sollen
- `2>`: Mit `2>` Ziel der Error-Ausgabe spezifizieren
- `error.txt`: Dateiname der Error-Ausgabe - hier die Datei „error.txt“

Bei längeren Pfaden oder Dateinamen, die ggf. Leerzeichen oder andere Zeichen enthalten, sind diese in Anführungszeichen zu setzen. Dies gilt sowohl für Paketnamen als auch für Parameter-Dateien.

## 4.2 Datenprüfprogramm

Das Datenprüfprogramm wurde vor dem Hintergrund der Anforderungen der Richtlinie zur einrichtungs- und sektorenübergreifenden Qualitätssicherung (Qesü-RL) implementiert, um bereits vor Ort beim Leistungserbringer eine von der Datenannahme-/Bundesauswertungsstelle implementierte Datenprüfung durchzuführen. Das Datenprüfprogramm bezieht die Plausibilitätsregeln direkt aus der Spezifikation und testet Daten vor deren Verschlüsselung.

### 4.2.1 Umfang der Prüfungen

Es werden zwei wesentliche Bereiche mit diesem Programm geprüft:

- Schema-Konformität (Struktur)  
Die XML-Datei wird dabei auf Konformität mit dem zugrunde liegenden Schema (XSD) überprüft.
- Regel-Konformität (Inhalte)  
Die XML-Datei wird dabei auf Einhaltung der Regeln (XSLT) überprüft.

Die eigentliche Prüfung erfolgt in der XML-Export-Datei. Deren Struktur ist der entsprechenden Dokumentation bzw. dem gültigen XML-Schema zu entnehmen.

Die inhaltliche Prüfung selbst erfolgt über ein XSLT-Stylesheet und einen XSLT-Prozessor. Das Datenprüfprogramm verwendet die freie Version eines XSLT-Version 2.0-kompatiblen Programms (XSLT2). Die Einbindung von XSLT-Stylesheet und XSLT-Prozessor erfolgt über ein Java-Programm. Prinzipiell kann jeder XSLT2-fähige XSLT-Prozessor für die Prüfung auf dieser Grundlage verwendet werden. Das Datenprüfprogramm stellt eine Referenzimplementierung dar.

### 4.2.2 Ausgangskontrolle vor Versand

Eine Schemavalidierung der Ausgangsdateien vor dem Versand der XML-Daten wird aus folgenden Gründen als notwendig erachtet:

- Sicherstellung der Datenintegrität nach der Verarbeitung der Daten
- Frühe Feststellung von Fehlerquellen in der eigenen Datenverarbeitung
- Entlastung des nachfolgenden Datenservices von nicht validen Daten
- Vermeidung des Versands von Daten, die gegen den Datenschutz verstoßen

Aus diesen Gründen wurde das Datenprüfprogramm um die Möglichkeit erweitert die XML-Dateien auch nach der Verschlüsselung der XML-Elemente auf Schemavalidität zu prüfen.

Der Leistungserbringer verwendet abhängig vom Datenfluss die Schemata zur Übergabe an die Datenannahmestelle (LE-DAS) unter:

```
interface_LE_DAS\
```

Die Datenannahmestelle verwendet das folgende Schema zur Übergabe an die Bundesauswertungsstelle (BAS):

```
interface_DAS_BAS\interface_DAS_BAS.xsd
```

Diese Schemata können einfach in den Config-Dateien im Parameter `<xsd_path>` entsprechend angegeben werden. Dazu müssen nur mehrere Config-Dateien verwendet werden (je eine

pro Schema). Des Weiteren besteht die Möglichkeit, beim Programmstart mit dem Parameter `-xsd-path` das Schema zu übergeben.

Diese Config-Dateien können entweder in verschiedenen Verzeichnissen abgelegt oder mit dem Parameter `-c` oder `--config` beim Start des Datenprüfprogramms über die Konsole angegeben werden. Da in diesem Fall keine inhaltliche Prüfung erfolgen soll (nur XSD, kein XSLT) muss zusätzlich der Parameter `--no-spez-val` angegeben werden:

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar -c
config_schema.xml --no-spez-val
```

Diese notwendige Ausgangskontrolle können die Softwareanbieter/Webportal-Entwickler (bzw. die Leistungserbringer) und die Datenannahmestellen unabhängig von dem Datenprüfprogramm realisieren, indem sie eine Schemavalidierung gegen die o.g. Schemata durchführen.

Für eine Schemavalidierung gibt es zahlreiche Tools und Bibliotheken für alle bekannten Programmiersprachen (<http://www.w3.org/XML/Schema>).

### 4.2.3 Programmaufruf

Das Datenprüfprogramm erzeugt eine Ausgabe/Output-Datei, die der Eingabe/Input-Datei entspricht, die jedoch um die Ergebnisse der Prüfungen erweitert wird. Die durchgeführten Prüfungen entsprechen einer Prüfung auf Dokumenten- und Vorgangsebene (Datensatzebene).

Das Datenprüfprogramm kann mehrere Dateien in einem Aufruf prüfen. Daher gibt es entsprechende Ordner für die Ein- und Ausgabedateien. Sollten diese Ordner nach der letzten Prüfung nicht geleert worden sein, so werden die Dateien des Eingabeordners erneut geprüft und der Ausgabeordner wird parallel mit Datum und Uhrzeit gesichert und ein neues leeres Ausgabeverzeichnis angelegt.

Die Prüfungen umfassen die Schemaprüfung und die Überprüfung der Feldinhalte (auch feldübergreifend).

#### Parameter `-c` oder `-config`

Die Steuerung der Funktionen erfolgt über eine Konfigurationsdatei, deren Dateipfad dem Programm beim Programmstart mit dem Parameter `-c` oder `--config` beim Programmaufruf übergeben werden kann.

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar -c
C:/konfiguration/config.xml
```

Wenn keine Konfigurationsdatei übergeben wird, wird die Datei `./config.xml` gesucht und geladen. Wenn diese Datei nicht gefunden wird, wird eine Standard-`config.xml`-Datei im Start-Order angelegt.

#### Parameter `--no-spez-val`

Mit diesem Parameter wird das Prüfskript ausgeschaltet.

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar --no-spez-val
```

### Parameter `--no-schema-val`

Mit diesem Parameter wird die Schemaprüfung ausgeschaltet.

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar --no-schema-val
```

### Batch-Dateien

Beim Datenprüfprogramm werden beispielhafte Batchdateien mitgeliefert:

- `datenpruefprogramm_schema.bat`  
Hier wird eine reine Schemaprüfung (Nur XSD) anhand einer Konfigurationsdatei „`config_schema.xml`“ durchgeführt.
- `datenpruefprogramm_<Datenfluss>_<Verfahrensart>.bat`  
Hier wird sowohl die Schemaprüfung (XSD) als auch die Inhaltliche Prüfung (XSLT) anhand einer Konfigurationsdatei „`config_<Datenfluss>_<Verfahrensart>.xml`“ durchgeführt.

---

### Beispiel einer Konfigurationsdatei:

```
<?xml version= "1.0 " encoding= "UTF-8 " standalone= "no " ?>
<config>
  <provider>
    <address>12345 Musterstadt</address>
    <email>max.mustermann@musterfirma.de</email>
    <fax>0123/456798</fax>
    <function>Softwarehersteller</function>
    <name>Mustermann</name>
    <phone>0123/456789</phone>
    <registration>1234567</registration>
  </provider>
  <gui>>false</gui>
  <input_path recursive= "true">input</input_path>
  <output_path>output</output_path>
  <xsd_path>xsd/interface_LE_WEICH/interface.xsd</xsd_path>
  <xsl_path>xsl/2017_DPP_EB_V01.aqxsl</xsl_path>
</config>
```

---

Die Konfigurationsdatei besteht aus den folgenden Bereichen:

#### Provider (Softwarehersteller/Webportal-Entwickler)

Im Element `<provider>` werden Daten benötigt, aus denen hervorgeht, wer das Prüfmodul einbindet und ausführt. In der Regel ist dies der Softwarehersteller. Zu beachten ist, dass die Auswahlmöglichkeit im Element `<function>` auf Softwarehersteller eingeschränkt ist. Die Elemente `<fax>`, `<phone>` und `<address>` sind optional, die anderen sind Pflichtelemente.

**Achtung**

In den Elementen `<fax>`, `<phone>` und `<address>` dürfen auf keinen Fall die Angaben des Leistungserbringers eingetragen werden!

---

**GUI (Konsole)**

Für ein vereinfachtes Debugging gibt es die Möglichkeit, eine Konsole mit detaillierten Programmausgaben während der Verarbeitung über das Element `<gui>` und den Wert `true` zu öffnen. Der Standard-Wert ist `false`.

**Input\_Path (Eingabeverzeichnis) – überschreibbar mit Parameter –input**

Im Element `<input_path>` kann der Eingabeordner für die zu überprüfenden Exportdateien festgelegt werden. Das Element ist optional. Ohne diesen Parameter ist der Ordner `<arbeitsverzeichnis>\input\` der Standard-Eingabe-Ordner. Es werden alle Dateien mit der Dateierweiterung `.xml` verarbeitet. Wenn das Attribut `recursive` auf `true` steht, werden auch alle entsprechenden Dateien in Unterordnern berücksichtigt. Die Standard-Einstellung von `recursive` ist `false`.

**Output\_Path (Ausgabeverzeichnis) – überschreibbar mit Parameter –output**

Im Element `<output_path>` kann der Ausgabeordner festgelegt werden. Das Element ist optional. Ohne diesen Parameter ist der Standard-Ausgabeordner `<arbeitsverzeichnis>\output\`. Der Dateiname der Ausgabedatei ist dabei gleich dem der Eingabedatei. Ein ggf. nicht vorhandener Ordner wird angelegt.

**XSD\_Path (Schemaordner) – überschreibbar mit Parameter --xsd-path**

Im Element `<xsd_path>` wird der Schemapfad gesetzt. Es wird dabei entweder auf ein Verzeichnis gezeigt, in dem genau eine Schemadatei erwartet wird, oder es wird direkt auf eine `xsd`-Datei gezeigt. Das Element ist optional. Ohne diesen Parameter wird das Schema im Verzeichnis `<arbeitsverzeichnis>\xsd\interface_LE_WEICH` gesucht. Eine Spezifikationskonforme Protokollierung kann vom Datenprüfprogramm sichergestellt werden, wenn die weiche Schemavariante verwendet wird.

Um nach einer Schemavalidierung der XML-Dateien, die Weiterverarbeitung und dementsprechend die spezifikationskonforme Protokollierung auf Datensatzebene weiterhin zu ermöglichen, wurden neben der harten Schemavariante ein weiches Schema für die Schnittstellen LE und DAS eingeführt. Diese weiche Variante wird ausschließlich mit dem Datenprüfprogramm verwendet (Abbildung 6). Eine detaillierte Übersicht über die Anwendung weicher Schemata mit dem Datenprüfprogramm sind im Abschnitt B 3.2 zu finden.

**Hinweis**

Neben den Leistungserbringern sind nur die LQS/LKG und die BAS (DAS) jene Stellen im Datenfluss, welche QS-Daten gemäß den G-BA-RLentschlüsseln dürfen und die weiche Variante benötigen.

---

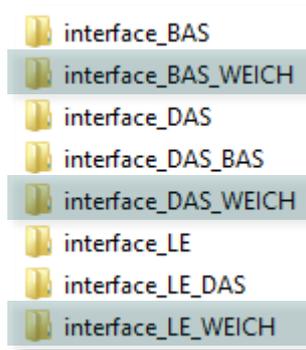


Abbildung 6: Weiche Schemavarianten für das DPP

#### **XSL\_Path (XSLT-Stylesheet-Ordner) – überschreibbar mit Parameter `--xsl-path`**

Im Element `<xsl_path>` kann der Quellordner für die XSLT-Stylesheets festgelegt werden. Das Element ist optional. Ohne diesen Parameter wird im Standard XSL-Ordner `<arbeitsverzeichnis>\xsl\` nach den XSLT-Stylesheets gesucht.

Mehrere Konfigurationsdateien können für dasselbe Datenprüfprogramm angelegt werden, um beispielsweise Dateien unterschiedlicher Spezifikationen zu validieren oder die Durchführung einer Eingangs- bzw. einer Ausgangskontrolle jeweils vor der Entschlüsselung und nach der Verschlüsselung zu ermöglichen.

#### **4.2.4 Verzeichnisstruktur**

Für das korrekte Funktionieren des Prüfprogramms ist neben den erforderlichen Dateien auch eine korrekte Verzeichnisstruktur notwendig.

In der config.xml wird der `<xsl_path>` definiert. Wenn der Parameter `<xsl_path>` auf ein Verzeichnis zeigt, muss in diesem Verzeichnis eine Stylesheet-Datei der folgenden Art vorliegen:

```
xsl\
```

Hierbei handelt es sich um ein kompiliertes Haupt-XSL-Stylesheet, welches die Prüfung entsprechend der Spezifikation durchführt. Es enthält alle Tests auf Regeln und Wertebereichsverletzungen. Ansonsten kann der Parameter `<xsl_path>` auf eine beliebige Stylesheet-Datei verweisen.

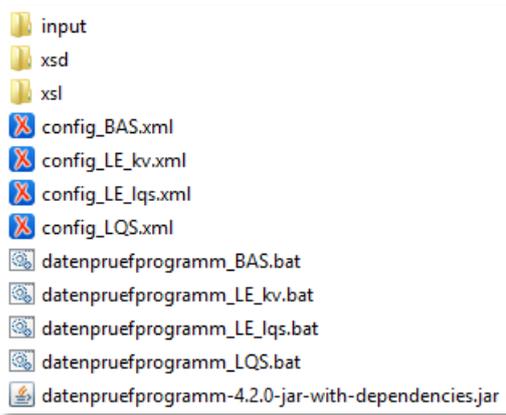


Abbildung 7: Verzeichnisstruktur eines DPP-Ordners

Die Verzeichnisstruktur kann über eine config.xml-Datei modifiziert werden. Ohne diese config.xml wird eine Standardkonfiguration angenommen, die das Prüfprogramm selbst in einer config.xml neu schreibt.

In Abbildung 7 ist eine typische Verzeichnisstruktur mit allen gemäß RL-Vorgaben sinnvollen config- und zugehörigen Batch-Dateien sowie dem Programm abgebildet. Innerhalb dieses Grundordners befinden sich die Verzeichnisse, welche in der Beschreibung der Konfiguration erläutert werden.

#### 4.2.5 Ausgabe

Nach dem Prüflauf wird ein Ordner `<output_path>` erzeugt, der die geprüften Dateien und deren Datenflussprotokolle beinhaltet.

##### **Geprüfte Dateien**

Im Ordner `<output_path>/files` liegen nun die geprüften Quell-Dateien, welche um das Ergebnis der Prüfung erweitert worden sind. Jeder Fall wird innerhalb der XML-Datei geprüft und in der XML-Struktur abgelegt. Zudem wird ein neuer Eintrag als `validation_provider` erzeugt.

##### **Protokolle**

Im Ordner `<output_path>/protocol` liegen die Datenflussprotokolle, die den entsprechenden Dateien im Ordner `<output_path>/files` entsprechen, in denen QS-Daten entfernt wurden.

##### **HTML-Protokolle**

Im Ordner `<output_path>/html` liegt eine `index.html`, in der auf vereinfachte Sichten der im Ordner `<output_path>/protocol` erstellten Protokolle verwiesen wird.

<a href="#">5d29aac7-d480-4377-ad2b-070596b27dda.xml</a>	5d29aac7-d480-4377-ad2b-070596b27dda	OK
<a href="#">65238192-1879-4fa2-84e6-a4323f52522c.xml</a>	65238192-1879-4fa2-84e6-a4323f52522c	OK
<a href="#">8e5e8197-5478-46b5-8fd3-cb527d0141a2.xml</a>	8e5e8197-5478-46b5-8fd3-cb527d0141a2	OK
<a href="#">9afe7686-7f73-43f8-a5ce-27d5139064f4.xml</a>	9afe7686-7f73-43f8-a5ce-27d5139064f4	OK
<a href="#">9d6359cf-2b70-4706-8670-5dd51b67306f.xml</a>	9d6359cf-2b70-4706-8670-5dd51b67306f	WARNING

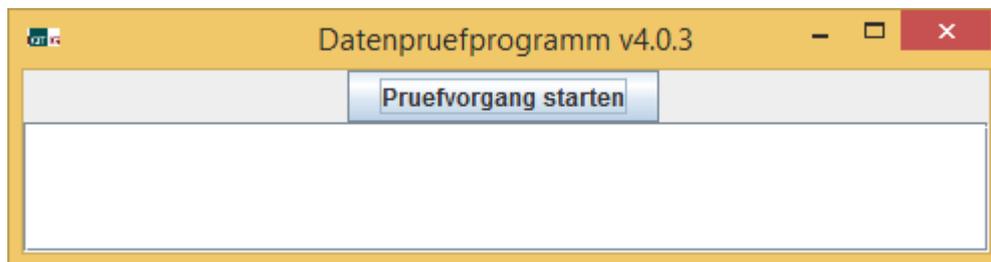
Abbildung 8: Beispiel für eine Index.html Datei im Ordner <output\_path>/html

Da für die Rückprotokollierung nur die Übertragung der unter <output\_path>/files abgelegten Datei spezifiziert ist, wird die Darstellung außerhalb von QS-Programmen beim Leistungserbringer durch ein eigenes Stylesheet ermöglicht, das sich an der Darstellung des Datenprüfprogramms orientiert. Die Dokumentation in Bezug auf dieses Stylesheet und dessen Einbindung sind Absatz „Lokale Transformation (Empfehlung)“ im Abschnitt „Die Rückprotokollierung“ der technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

#### 4.2.6 Grafische Oberfläche

Wird der Parameter GUI in der Konfigurationsdatei auf „true“ gesetzt, wird das Datenprüfprogramm mit einer einfachen grafischen Oberfläche gestartet.

Abbildung 9: Grafische Oberfläche des Datenprüfprogramms



Bei der grafischen Oberfläche muss zum Starten die Schaltfläche „Pruefvorgang starten“ gedrückt werden.

Die grafische Oberfläche zeigt die Ausgabe im Fensterbereich direkt an. Am Inhalt des Ausgabe-Ordners ändert sich nichts; beide Laufvarianten (grafische Ausgabe oder Konsolenausgabe) erzeugen den gleichen Output.

#### 4.2.7 Programmierschnittstelle – API

Sämtliche Funktionen des Datenprüfprogramms können über eine Programmierschnittstelle (API) aufgerufen und direkt in einem Java-Umfeld verwendet werden.

### 4.3 Verschlüsselungspaket

Das IQTIG stellt ein Ver- und Entschlüsselungspaket für die Anwendung in der externen Qualitätssicherung entsprechend QSKH-RL und Qesü-RL zur Verfügung. Die Ver- und Entschlüsse-

lungsalgorithmen sind entsprechend der im Abschnitt „Gesicherte QS-Datenübertragung“ beschriebenen W3C-Standards<sup>57</sup> implementiert. Das bedeutet, dass an jeder beliebigen Stelle im Workflow auch jede andere Implementierung, die sich an diese Standards hält, verwendet werden kann.

Das Verschlüsselungspaket besteht aus dem TPACKER für die Transportverschlüsselung und dem XPACKER für die XML-Verschlüsselung. Für alle Anwender, die händisch die Verschlüsselung durchführen müssen, stellt der GPACKER mit seiner grafischen Oberfläche eine interaktive Alternative zur Verwendung der Programme XPACKER und TPACKER dar.

#### 4.3.1 XPACKER – XML-Verschlüsselung

Das Programm XPACKER.jar dient zur Ver- und Entschlüsselung einzelner XML-Elemente innerhalb einer XML-Datei. Bevor diese verschlüsselt werden, wird jedes einzelne Element mit „base64“ komprimiert bzw. „gezippt“ und beim Entschlüsseln parallel wieder „entpackt“.

Eine weitere Funktion des Programms dient zur Generierung eines asymmetrischen Schlüssel-paares (privater und öffentlicher Schlüssel).

Die Verschlüsselung wird mit dem hybriden Verfahren (siehe Absatz „Die Verschlüsselung“ im Abschnitt „Gesicherte Datenübertragung“ der technischen Dokumentation zur Basisspezifikation für Leistungserbringer):

- Die Verschlüsselung der Daten erfolgt mit einem AES-128bit-Schlüssel im CBC-Mode ([aes128-cbc](#)).
- Die Verschlüsselung des symmetrischen Schlüssels erfolgt mit einem RSA-Schlüssel im OAEP-Mode ([rsa-oaep-mgf1p](#)).

Weitere technische Details beschreibt der W3C-Standard „XML Encryption Syntax and Processing“.<sup>58</sup>

Die folgende Abbildung zeigt ein XML-Dokument nach der Verschlüsselung eines XML-Elements (QS-Daten):

---

<sup>57</sup> RSA2048: <http://www.w3.org/TR/xmlenc-core/>

AES128: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

<sup>58</sup> Ebd.

```

<encryption>
  <ds:KeyInfo Id="Pub_key_Bundesauswertungsstelle.pub-qs_data-2.2.0">
    <xenc:EncryptedKey Id="qs_data" xmlns:xenc="http://www.w3.org/2001/04/xmenc#">
      <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmenc#rsa-oaep-mgf1p"/>
      <xenc:CipherData>
        <xenc:CipherValue>dybYkkepEipN48IE0mbj28aw83HDyMqJQxvT3nSof34SdLHLpZ+A==</xenc:CipherValue>
      </xenc:CipherData>
    </xenc:EncryptedKey>
  </ds:KeyInfo>
</encryption>
</header>
<body>
  <data_container>
    <care_provider>
      <cases module="KAT_FU_A" pseud_procedure="KAT">
        <case>
          <patient>
            <case_admin>
              <qs_data module="KAT_FU_A" xsi:type="qs_data_kat_fu_a_type">
                <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmenc#Content" xmlns:xenc="http://www.w3.org/2001/04/xmenc#">
                  <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmenc#aes128-cbc"/>
                  <xenc:CipherData>
                    <xenc:CipherValue>xEO1RE59SeMKaRs1V4DlZ2QJDDV</xenc:CipherValue>
                  </xenc:CipherData>
                </xenc:EncryptedData>
              </qs_data>
            </case_admin>
          </patient>
        </case>
      </cases>
    </care_provider>
  </data_container>
</body>

```

Abbildung 10: Verschlüsselung eines XML-Elements (qs\_data)

In diesem XML-Dokument existieren einige spezielle Elemente. Die wichtigsten sind:

- EncryptedData ist das einschließende Element für die XML-Verschlüsselung. Der gesamte Inhalt des übergeordneten Elements einschließlich der Attribute ist verschlüsselt.
- CipherData ist das verschlüsselte Element.
- CipherValue enthält die verschlüsselten Daten.
- KeyInfo enthält Informationen über den Schlüssel, mit dem die Daten verschlüsselt worden sind.
- Das Attribut id enthält den Namen des PublicKeys und den Namen des zu verschlüsselnden „Tags“ und die aktuelle Versionsnummer des jeweils aktuellen XPackers.

### Syntax/Hilfe

Mit dem Befehl `-h` wird die Syntax des Programms und eine Parameterübersicht ausgegeben:

```
java -jar XPacker-4.1.2-jar-with-dependencies.jar -h
```

### Verschlüsseln

Die Verschlüsselung wird durch den Parameter `-e` aktiviert.

Benötigt wird der Parameter `-k`, gefolgt von Dateinamen des öffentlichen Schlüssels, der Parameter `-t` mit durch Komma getrennten Element-Namen sowie der Parameter `-2` mit Tag-Name, in den der verschlüsselte Schlüssel aufgenommen wird. Optional sind die Parameter `-f` und `-o`.

`-f` gibt die einzulesende XML-Dateien an und `-o` die auszugebende Datei. Falls diese Parameter fehlen, wird die Standard Ein-/Ausgabe verwendet.

---

**Beispiel:**

```
java -jar XPacker-4.1.2-jar-with-dependencies.jar -e -f in-  
file.xml -o outfile.xml  
-k datenannahme.pub  
-t qs_data -2 encryption
```

---

**Entschlüsseln**

Die Entschlüsselung entspricht dem Verschlüsseln, nur dass der Parameter `-2` entfällt und bei dem Parameter `-k` der private Schlüssel angegeben werden muss und der Parameter `-e` durch `-d` ersetzt wird.

---

**Beispiel:**

```
java -jar XPacker-4.1.2-jar-with-dependencies.jar -d -f in-  
file.xml -o outfile.xml  
-k datenannahme.pri -t qs_data
```

---

**Schlüsselpaare erzeugen**

Für die Erzeugung eines Schlüsselpaares wird der Parameter `-g` verwendet. Optional kann mit `-o` ein Dateiname angegeben werden. An diesen Namen wird `.pub` für den öffentlichen Schlüssel und `.pri` für den privaten Schlüssel angehängt. Falls der Parameter `-o` fehlt, wird im aktuellen Verzeichnis ein `key.pub` für den öffentlichen Schlüssel und ein `key.pri` für den privaten Schlüssel erzeugt.

---

**Beispiel:**

```
java -jar XPacker-4.1.2-jar-with-dependencies.jar -g -k da-  
tenannahme
```

Erzeugt im aktuellen Verzeichnis die Dateien `datenannahme.pub` und `datenannahme.pri`.

---

**4.3.2 TPacker – Transportverschlüsselung**

Das Programm `TPacker.jar` dient zur Transportverschlüsselung. Neben dem Ver- und Entschlüsseln werden die Dateien auch ge- und entpackt (`.zip`).

Beim Ver- und Entschlüsseln wird aus dem symmetrischen Schlüssel (Passwort) über ein MD5-Hash ein AES-Schlüssel erzeugt. Die eigentliche Verschlüsselung erfolgt im ECB-Mode. Als Padding<sup>59</sup> verwendet das Programm den PKCS5-Standard.

Die Steuerung der Funktionen erfolgt über Parameter beim Programmaufruf.

**Syntax/Hilfe**

Mit dem Befehl `-h` werden die Syntax des Programms und eine Parameterübersicht ausgegeben

---

<sup>59</sup> Als „Padding“ wird hier das Auffüllen eines unvollständigen Blocks am Ende des Datenstroms bezeichnet.

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar -h
```

### Isolierte Varianten

Das Packen und Verschlüsseln bzw. das Entschlüsseln und Entpacken können unabhängig voneinander durchgeführt werden.



#### Achtung

#### Anwender- und Übertragungsfehler

Die im Folgenden beschriebenen Aufrufe von isoliertem Ver- und Entschlüsseln sind komplex und können schnell zu Anwender- und Übertragungsfehlern führen.

Es müssen die richtigen Schritte mit den richtigen Parametern exakt in der richtigen Reihenfolge durchgeführt werden. Dazu sind die im Folgenden beschriebenen Varianten zu nutzen.

---

### Verschlüsseln

Die Verschlüsselung wird durch die Parameter `-e` aktiviert. Benötigt wird noch der Parameter `-f` mit dem Dateinamen der zu verschlüsselnden Datei und der Parameter `-p` mit dem Passwort.

Falls nichts anderes angegeben wird, entspricht der Name der Ausgabedatei dem mit dem Parameter `-f` angegebenen Eingabedateinamen.

Mit dem Parameter `-o` kann man den Namen der Ausgabedatei ändern. In beiden Fällen wird an den Dateinamen ein Zeitstempel und `.aes` hinzugefügt. Falls der Zeitstempel nicht gewünscht ist, wird dieser mit dem Parameter `-t` deaktiviert.

---

#### Beispiel:

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar -e -f Datei  
-o Dateiname -p geheim
```

würde die Datei `Dateiname_2012_01_01_121212.aes` erzeugen, wenn das Programm am 01.01.2012 um 12:12 Uhr und 12 Sekunden ausgeführt wird. Die Verschlüsselung würde mit dem Schlüssel „privat“ durchgeführt.

---

### Entschlüsseln

Beim Entschlüsseln geht man genau wie beim Verschlüsseln vor, nur dass man den Parameter `-e` durch den Parameter `-d` ersetzt.

Der Dateiname der Ausgabedatei entspricht der von Parameter `-f`. Falls ein Suffix `.aes` enthalten ist, wird dieses entfernt.

---

#### Beispiel:

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar -t -d -f Datei.aes -p geheim
```

würde die Datei „Datei“ erzeugen.

---

**Packen**

Beim Packen geht man genau wie beim Verschlüsseln vor, nur dass man die Parameter `-e` durch `-z` ersetzt und der Parameter `-p` überflüssig ist. Das Suffix der gebildeten Datei lautet `.zip`.

Der Parameter `-f` kann eine oder mehrere, durch Komma(,) getrennte Dateinamen (ohne Leerzeichen) enthalten. Bei Angabe eines Verzeichnisses gibt es einen Fehler.

---

**Beispiel:**

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar -t -z -f Datei -o Dateiname
```

würde die Datei „Dateiname.zip“ erzeugen.

---

**Entpacken**

Beim Entpacken geht man genau wie beim Packen vor, nur dass man den Parameter `-z` durch den Parameter `-u` ersetzt.

Die Ausgabe über Parameter `-o` ist immer ein Verzeichnis. Falls kein Verzeichnis angegeben wird, wird das aktuelle Verzeichnis als Ausgabeverzeichnis verwendet.

---

**Beispiel:**

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar -t -u -f Datei.zip -o Verzeichnis
```

würde die Datei „Datei(en)“ im Verzeichnis „Verzeichnis“ entpacken.

---

**Integrierte Varianten**

Das Packen und Verschlüsseln bzw. das Entschlüsseln und Entpacken kann in einem Schritt zusammengefasst werden.

**Verschlüsseln mit integriertem Verpacken**

Es ist auch möglich, in einem Schritt zu packen und zu verschlüsseln. Hier kombiniert man die Parameter zu `-ze`. Es wird immer zuerst gepackt und dann verschlüsselt. Als Suffix wird dann `.zip.aes` angehängt.

---

**Beispiel:**

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar -t -ze -f Datei -o Dateiname -p geheim
```

würde die Datei „Dateiname.zip.aes“ erzeugen.

---

**Entschlüsseln mit integriertem Entpacken**

So wie das Packen und Verschlüsseln, kann auch das Entschlüsseln und Entpacken in einem Schritt zusammengefasst werden. Hier kombiniert man die Parameter zu `-ud`. Es wird immer zuerst entschlüsselt und dann entpackt. Die Ausgabe erfolgt auch hier immer in ein Verzeichnis.

---

**Beispiel:**

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar -t -ud -f Datei.zip.aes -p geheim
```

würde die Datei „Datei.zip.aes“ im aktuellen Verzeichnis entschlüsseln und entpacken.

---

**Der Modus „unsafe“**

Der TPacker ist standardmäßig so eingestellt, dass vor der Verschlüsselung das zu verschlüsselnde Dokument darauf untersucht wird, ob ein Element VERSICHERTENID bzw. VERSICHERTENIDNEU vorhanden ist. Falls diese Prüfung positiv ausfällt oder kein wohlgeformtes XML gefunden wird, unterbricht eine entsprechende Fehlermeldung die Verschlüsselung. Durch diese Maßnahme soll verhindert werden, dass versehentlich eine Transportdatei VOR der XML-Verschlüsselung erzeugt werden kann.

Um diesen Mechanismus auszuschalten, kann der Parameter

```
--unsafe
```

gemeinsam mit dem Verschlüsselungsparameter -e verwendet werden.

---

**Beispiel:**

```
java -jar TPacker-4.1.2-jar-with-dependencies.jar --unsafe -e -f Datei -o Dateiname -p geheim
```

würde beim Erzeugen der verschlüsselten Datei keine Prüfung auf den Inhalt vornehmen.

---

### 4.3.3 TPacker und XPacker ohne externe Abhängigkeiten

XPacker und TPacker werden vom IQTIG bei Bedarf ohne externe Abhängigkeiten (ohne fremde Bibliotheken) zur Verfügung gestellt. Die aktuelle Version ist auf der Kommunikationsplattform des IQTIG zu finden.

---

**Hinweis**

Die Programmvarianten ohne externe Abhängigkeiten sind nur für die Softwareanbieter relevant, die selber die Programme in ein JAVA-Umfeld integrieren und dabei Konflikte mit bereits installierten Bibliotheken vermeiden wollen.

---

### 4.3.4 Programmierschnittstelle – API

Sämtliche Funktionen der Verschlüsselungsprogramme können über je eine Programmierschnittstelle (API) aufgerufen und direkt in einem Java-Umfeld verwendet werden.

### 4.3.5 GParser

Für alle Anwender, die händisch die Verschlüsselung durchführen müssen, stellt der GParser mit seiner grafischen Oberfläche eine interaktive Alternative zur Verwendung der Programme XPacker und TPacker dar. Der GParser integriert dabei den XPacker und TPacker über die GUI, die

als einzelne Programme damit nicht benötigt werden. Er ermöglicht die Verschlüsselung und Komprimierung von QS-Dateien im XML-Format, die im Rahmen der Qesü-RL oder QSKH-RL erstellt worden sind. Eine ausführliche Dokumentation liegt der Komponente bei.

#### **4.4 LE-Pseudonymisierungsprogramm**

Das LE-Pseudonymisierungsprogramm richtet sich ausschließlich an die Datenannahmestellen (DAS), um die leistungserbringeridentifizierenden Daten bei den Qesü-Modulen vor der Weiterleitung an das IQTIG nach einem einheitlichen Verfahren zu pseudonymisieren. Alle weiteren Informationen und die Beschreibung zur LE-Pseudonymisierung sind dem ZIP-Archiv und den darin enthaltenen Dokumentationen (Anwenderhandbuch und Datenflussspezifikation) auf der IQTIG-Website zu entnehmen.

##### **Die Dokumentation besteht aus:**

- Anwenderhandbuch mit der Beschreibung der einzelnen Programmkomponenten (grafische Benutzeroberfläche, Konsolenanwendung, Java-API) sowie die Funktionenbeschreibung (Konfiguration, Pseudonymisierung, Depseudonymisierung, Datenübertragung etc.).
- Datenflussspezifikation mit der Beschreibung der konkreten Datenflüsse bei der Pseudonymisierung und Datenübertragung der Zusatzinformationen.

# C Anhang

## Glossar

Begriff	Beschreibung
Abrechnungsdaten	Daten, die von Leistungserbringern über von ihnen erbrachte Leistungen zum Zweck der Abrechnung mit den Kostenträgern dokumentiert und übermittelt wurden.
Anonymisierung	Verändern personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbar natürlichen Person zugeordnet werden können. (§ 3 Abs. 6a BDSG)
Auslösekriterien	Algorithmus zur Auslösung der Dokumentationspflicht (QS-Filter).
Basisspezifikation	Gemeinsame Spezifikation für stationäre und sektorenübergreifende QS-Dokumentationen.
Bundesdatenpool	Zusammenführung aller bundesweit dokumentierten QS-Daten.
Datenannahmestellen	Stellen, an die die Leistungserbringer oder andere Daten liefernde Stellen (z.B. Krankenkassen) die erhobenen Daten übermitteln. Sie prüfen die übermittelten Daten auf Plausibilität, Vollständigkeit und Vollzähligkeit (§ 9 Qesü-RL).
Datenbasis	Im Einzelfall festzulegende bzw. festgelegte Gesamtmenge von auszuwertenden bzw. ausgewerteten Daten.
Datenfeld	Kleinste Einheit eines Datensatzes (z. B. Angabe des Geschlechts im Dokumentationsbogen).
Datenfluss	Übermittlung der Daten der QS-Verfahren in einem festgelegten Format und Inhalt, die vom Leistungserbringer über eine Datenannahmestelle, Vertrauensstelle (nur sektorenübergreifende PID-Verfahren) bis zur Datenauswertungsstelle gelangen. Die Datenflüsse sind grundsätzlich in der QSKH-RL und der Qesü-RL des G-BA festgelegt.
Datensatz	Eine zusammenhängende Menge von QS-Daten, die einem Fall (beispielsweise einem Patienten) zugeordnet wird.
Datenvalidierung	Verfahren zur Überprüfung von QS-Daten einerseits auf Vollzähligkeit, Vollständigkeit und Plausibilität (statistische Basisprüfung), andererseits ihre Übereinstimmung (Konkordanz) mit einer Referenzquelle wie bspw. der Krankenakte (Stichprobenverfahren mit Datenabgleich).

<b>Begriff</b>	<b>Beschreibung</b>
Diagnosis Related Groups (DRG)	Diagnosebezogene Fallgruppen zur pauschalierten Abrechnung stationärer Behandlungsfälle. Wesentliche Grundlage für die Ermittlung einer DRG, das sogenannte Grouping, sind Diagnosen- (ICD) und Prozeduren-Kodes (OPS).
Dokumentation	siehe: QS-Dokumentation
Einheitlicher Bewertungsmaßstab (EBM)	Vergütungssystem für die Abrechnung von Leistungen im vertragsärztlichen Bereich.
Einrichtung	siehe: Leistungserbringer
Erfassungsjahr	Das Jahr, in dem die Daten erhoben werden. Hierauf beruhen die Ergebnisse der Indikatoren. Die Kriterien für die Abgrenzung des Erfassungsjahres sind in der Spezifikation zur Sollstatistik definiert.
Follow-up	auch: Mehrpunktmessung, Mehrzeitpunktmessung. Versorgungsleistungen, deren Qualität (z.B. ein Behandlungserfolg) nicht einmalig, sondern zu unterschiedlichen weiteren Zeitpunkten gemessen wird. Die Messungen können in einem bestimmten oder auch in unterschiedlichen Sektoren (vgl. QSKH-RL) stattfinden.
Gemeinsamer Bundesausschuss (G-BA)	Oberstes Beschlussgremium der gemeinsamen Selbstverwaltung der Ärzte, Zahnärzte, Psychotherapeuten, Krankenhäuser und Krankenkassen in Deutschland. Er bestimmt in Form von Richtlinien den Leistungskatalog der Gesetzlichen Krankenversicherung (GKV) für etwa 70 Millionen Versicherte und legt damit fest, welche Leistungen der medizinischen Versorgung von der GKV erstattet werden.
Indikator	„Anzeiger“. Auch: Qualitätsindikator. Quantitatives Maß zur Bewertung der Erreichung eines Qualitätsziels einer bestimmten gesundheitlichen Versorgungsleistung.
Kostenträger	Personen und Institutionen, die die Kosten für medizinische Versorgungsleistungen tragen. Im Rahmen der gesetzlich verankerten Qualitätssicherung sind dies die gesetzlichen Krankenkassen.
Leistungserbringer	Personen und Einrichtungen, die medizinische Versorgungsleistungen erbringen bzw. bereitstellen. Der Begriff wird im SGB V auch für Ärzte und ärztliche Einrichtungen sowie für zugelassene Krankenhäuser gem. § 108 SGB V genutzt.
Leistungserbringeridentifizierende Daten (LID)	Daten, die eindeutig einen bestimmten Leistungserbringer identifizieren (z.B. Institutionskennzeichen oder Betriebsstättennummer).
Missing Values	„Fehlende Werte“, z.B. fehlende Antworten und nicht auswertbare Antworten bei der Auswertung eines Fragebogens.

Begriff	Beschreibung
Operationen- und Prozedurenschlüssel (OPS)	Kodierungssystem für medizinische Leistungen bzw. verbindlicher Abrechnungsschlüssel zwischen medizinischen Leistungserbringern und Kostenträgern (Gesetzliche Krankenkassen).
Patientenidentifizierende Daten (PID)	Daten, die eindeutig einen bestimmten Versicherten identifizieren (z.B. Versichertennummer).
Plausibilitätsprüfung	Statistisches Verfahren, mit dem die Dokumentationsdaten auf erlaubte und/oder fehlende Werte, Widerspruchsfreiheit, Werteverteilung und bekannte Korrelationen geprüft werden.
Probetrieb	Erprobung eines QS-Verfahrens in einer begrenzten Anzahl von Einrichtungen. Ziel ist die Prüfung, ob die für das QS-Verfahren benötigten Daten entsprechend der vom Auftragnehmer vorgesehenen Planungen für die vorgelegten Indikatoren und Instrumente erhebbar und die Ergebnisse umsetzbar sowie für die Durchführung der Qualitätssicherung verwertbar sind.
Pseudocode	Programmcode, der das zugrunde liegende Prinzip eines Algorithmus beschreibt, selbst aber nicht lauffähig ist. Er dient zur Veranschaulichung, unabhängig von der konkret zu verwendenden Programmiersprache.
Pseudonymisierung	Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren (§3 Abs. 6a BDSG).
QS-Auslösung	Initiierung einer Dokumentationspflicht zu Zwecken der Qualitätssicherung (QS-Dokumentation). Bei einer Erhebung vorhandener Daten (z.B. Sozialdaten bei den Krankenkassen) analog das Kriterium, das die Lieferung eines bestimmten Datensatzes auslöst.
QS-Daten	Sammelbegriff für alle Daten, die im Zuge eines QS-Verfahrens erhoben und ausgewertet werden.
QS-Dokumentation	Gesonderte Erhebungen der Leistungserbringer zu Diagnose- und Behandlungsdaten der Patienten durch die Leistungserbringer für die Qualitätssicherung.
QS-Filter	Algorithmus, der auf Grundlage festgelegter Kriterien die für die Qualitätssicherung durch die Leistungserbringer zu dokumentierenden Patienten und deren Daten „filtert“. Die Kriterien hierzu werden in einer Spezifikation definiert.
QS-Filter-Software	Implementierung der Spezifikation für den QS-Filter.
QS-Verfahren	hier: QS-Verfahren der gesetzlichen Qualitätssicherung. Medizinischer Themen- oder Behandlungsbereich, der im Rahmen der bundesweiten Qualitätssicherung dokumentationspflichtig ist.

Begriff	Beschreibung
Qualität	Bezogen auf die Gesundheitsversorgung: Grad, in dem versorgungsrelevante Ergebnisse, Prozesse und Strukturen bestimmte, definierte Anforderungen erfüllen.
Qualitätsindikator	siehe: Indikator
Qualitätssicherung	Sammelbegriff für unterschiedliche Ansätze und Maßnahmen zur Sicherstellung festgelegter Qualitätsanforderungen bzw. zur Erreichung bestimmter Qualitätsziele. Hier: Gesetzliche Qualitätssicherung im Gesundheitswesen nach §§ 135-139 SGB V.
Qualitätssicherung, externe stationäre	Einrichtungübergreifende Qualitätssicherung für medizinisch-pflegerische Leistungen, die ausschließlich im stationären Sektor erbracht werden.
Qualitätssicherungsmaßnahmen	Strukturierte, in Richtlinien geregelte Vorgehensweise, die Leistungserbringer bei der kontinuierlichen Qualitätsverbesserung unterstützt. Auslöser der Qualitätssicherungsmaßnahmen sind rechnerische Auffälligkeiten im Ergebnis eines Qualitätsindikators.
Regelbetrieb	auch: Routinebetrieb oder Echtbetrieb. Verpflichtende und flächendeckende Umsetzung eines QS-Verfahrens.
Risikoadjustierung	Methode zur Ermittlung und Gewichtung wesentlicher Einflussfaktoren (individueller Risiken) auf die Ausprägung einer Messgröße. Eine Risikoadjustierung ist vor allem bei Qualitätsindikatoren erforderlich, die sich auf Behandlungsergebnisse und einen Einrichtungsvergleich beziehen. Um hier einen fairen Vergleich zu erhalten, dürfen nur in gleichem Umfang erkrankte Patienten in Einrichtungen mit einer vergleichbaren Klientel miteinander verglichen werden.
Routinedaten	hier: Daten, die wesentlich zur Abwicklung von Geschäfts- und Verwaltungsabläufen erhoben werden (z.B. Abrechnungsdaten, personenbezogene administrative Daten). Abseits des uneinheitlichen Sprachgebrauchs stehen die Sozialdaten bei den Gesetzlichen Krankenkassen (auch: GKV-Routinedaten) im Vordergrund des Interesses, da sie gem. § 299 Abs. 1a SGB V zu Zwecken der Qualitätssicherung verwendet werden dürfen. Diese beinhalten insbesondere die abrechnungsrelevanten Daten für ambulante und stationäre Versorgungsleistungen (§§295 und 301 SGB V), für Arznei-, Heil- und Hilfsmittel (§§ 300 und 302 SGB V) sowie die Versichertenstammdaten (§ 284 SGB V).
Sektor	Institutionell, d.h. durch unterschiedliche Abrechnungsmodalitäten getrennte Bereiche der medizinisch-therapeutischen Versorgung im deutschen Gesundheitswesen (z.B. ambulant/stationär).

Begriff	Beschreibung
Sollstatistik	Aufstellung der im Erfassungsjahr zu dokumentierenden Leistungen, die vom Krankenhaus durch Konformitätserklärung schriftlich bestätigt wird. Sie gibt die zu erwartende Anzahl von Fällen in den einzelnen QS-Verfahren (Soll) an und bildet zusammen mit der Anzahl der tatsächlich durch die Leistungserbringer gelieferten Daten (Ist) die Grundlage der Vollzähligkeitsprüfung.
Sozialdaten	Einzelangaben über die persönlichen und sachlichen Verhältnisse (personenbezogene Daten), die von den sozialrechtlichen Leistungsträgern zur Erfüllung ihrer gesetzlichen Aufgaben gesammelt und gespeichert werden.
Spezifikation	Datensatzbeschreibung. Festlegung, welche Daten für die Qualitätssicherung erhoben bzw. übermittelt werden müssen, welche Prüfalgorithmen zur Anwendung kommen (z.B. für Plausibilitätsprüfungen) und wie die QS-Auslösung operationalisiert ist. Im Rahmen der Neuentwicklung von QS-Verfahren ist die Spezifikation als das Ergebnis der informationstechnischen Aufbereitung zu betrachten.
Strukturierter Dialog	Instrument der Qualitätsförderung. Strukturiertes Verfahren, das Einrichtungen bei der kontinuierlichen Verbesserung von Prozessen und Qualität unterstützt. Auslöser des Strukturierten Dialogs sind rechnerische Auffälligkeiten im Ergebnis eines Qualitätsindikators im Rahmen der statistischen Basisprüfung der übermittelten QS-Daten (QSKH-RL). Abweichend von der Definition in der QSKH-RL bezeichnet der Begriff „Strukturierter Dialog“ in der Qesü-RL nur den Prozessschritt, der dem schriftlichen Teil der Durchführung von QS-Maßnahmen folgt.
Systempflege	Routinemäßige und kontinuierliche Evaluation und Anpassung der Qualitätsindikatoren, der Softwarespezifikation usw.
Verfahren	siehe: QS-Verfahren
Vertrauensstelle	Institution, die im Rahmen der sektorenübergreifenden Qualitätssicherung erhobene patientenidentifizierende Daten pseudonymisiert. Näheres regeln die Qesü-RL (§ 11), QSKH-RL (Anl. 2, § 4) sowie themenspezifische Bestimmungen.
Verweildauer	Dauer des stationären Aufenthalts eines Patienten, Abstand zwischen Aufnahme- und Entlassungsdatum.
Vollständigkeit	Erfassung aller zu einem einzelnen Behandlungsfall erforderlichen Angaben (Daten).
Vollzähligkeit	Erfassung aller dokumentationspflichtigen Behandlungsfälle zu einer bestimmten Versorgungsleistung.
Vollzähligkeitsprüfung	Abgleich der laut QS-Filter-Software zu erwartenden Anzahl von Fällen in einem QS-Verfahren (Soll) mit der Anzahl der tatsächlich durch die Leistungserbringer gelieferten Daten (Ist).

<b>Begriff</b>	<b>Beschreibung</b>
XSLT	Extensible Stylesheet Language Transformations. Programmiersprache zur Transformation von XML-Dokumenten in andere XML-Dokumente oder andere Dokumentformate wie HTML. Im QS-Kontext kann es auch für Datenprüfung und Protokollerstellung verwendet werden.