

Softwarebibliothek CORE.connect für JAVA

Nutzerdokumentation

Version: 6

Stand: 26.03.2007

Status: vorläufig

Kontakt: eSTATISTIK.core@destatis.de

© Statistisches Bundesamt Wiesbaden, Deutschland

Inhalt

1	Einleitung.....	3
2	Installation und Deinstallation.....	4
2.1	Systemvoraussetzungen	4
2.2	Installation	4
2.3	Deinstallation	4
3	Funktionsumfang.....	5
4	Überblick über die Java-Pakete und -Klassen.....	6
4.1	Paket <code>de.destatis.core.connect</code>	7
4.2	Paket <code>de.destatis.core.connect.document</code>	9
4.3	Paket <code>de.destatis.core.inspector</code>	10
5	Konfigurationseinstellungen	11
6	Statuscodes des zentralen Online-Dateneingangs.....	15
7	Fehlerschlüssel und -meldungen des Inspector-Moduls	16
7.1	Prüfebene: Syntax.....	17
7.2	Prüfebene: Semantik.....	17
7.3	Prüfebene: Autorisierung.....	18
7.4	Prüfebene: Daten	18
8	Weitere Dokumentation	21
8.1	Beispiele.....	21
8.2	Weitere Informationen	21

Änderungsverlauf

Version	Datum	Autor	Änderung
1	16.03.04	Bösnecker	Neuerstellung
2	06.09.04	Bösnecker	Anpassung wegen C/C++-Schnittstelle
3	07.03.05	Bösnecker	Anpassung an 1.0
	02.03.06	Bösnecker	Anpassung an 1.03
4	03.03.06	Bösnecker	Anpassung an 1.1
5	08.01.07	Bösnecker	Erweiterungen in Kap. 3 und 8, Kap. 4, 5, 6, 7 neu
	01.03.07	Bösnecker	Neuer Fehlerschlüssel 43022 in Kap. 7.4
6	26.03.07	Bösnecker	Anpassungen an 1.2

1 Einleitung

Die Softwarebibliothek CORE.connect unterstützt Softwarehersteller und Befragte bei der Realisierung von Statistikmodulen für die Erstellung, die Prüfung und den Versand von Datenlieferungen im XML-Format DatML/RAW an den zentralen Online-Dateneingang der Statistischen Ämter.

Ein Statistikmodul ist eine in die Unternehmenssoftware integrierte Softwarekomponente, die die Gewinnung der statistischen Daten aus der DV des Befragten durchführt oder steuert und aus den so gewonnenen Daten die Datenlieferung erstellt und versendet. Auf diese Weise lässt sich das gesamte Datenerhebungsverfahren beim Befragten automatisieren (s. Abb. 1).

Die Softwarebibliothek bietet alle notwendigen Funktionen, um eine Datenlieferung im XML-Format DatML/RAW zusammenzustellen und an den zentralen Online-Dateneingang zu versenden. Durch die maschinelle Erstellung und Prüfung der Datenlieferung werden die Aufwände bei den Befragten erheblich verringert und gleichzeitig die Qualität der statistischen Rohdaten deutlich verbessert. Die Softwarebibliothek wird kostenlos von den statistischen Ämtern zur Verfügung gestellt.

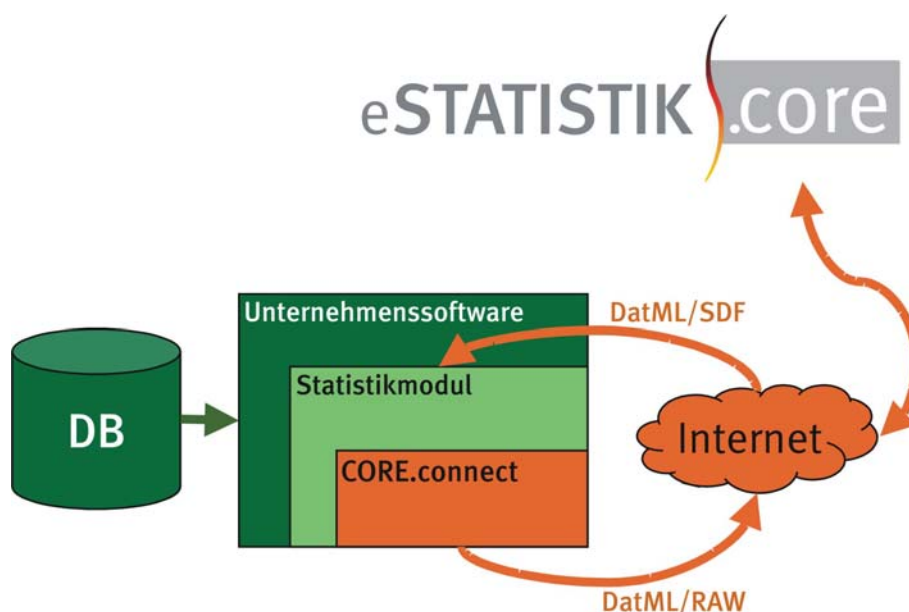


Abbildung 1: Verwendung der Softwarebibliothek CORE.connect

Die Softwarebibliothek CORE.connect richtet sich an DV-Experten bei den Befragten und bei Softwareherstellern, die die Erstellung und den Versand von Datenlieferungen in DatML/RAW in ihre Datenverarbeitung bzw. Softwareprodukte integrieren wollen. Für die Nutzung werden dementsprechend einschlägige DV-Kenntnisse vorausgesetzt.

Referenzdokumente

- [SPEZ] Spezifikation und Dokumenttypdefinition von DatML/RAW und DatML/SDF
<http://www.statistik-portal.de> → Online-Erhebungen → CORE → Download
- [KOMM] Kommunikationsschnittstelle des zentralen Online-Dateneingangs von eSTATISTIK.core
<http://www.statistik-portal.de> → Online-Erhebungen → CORE → Download

2 Installation und Deinstallation

2.1 Systemvoraussetzungen

Die Softwarebibliothek CORE.connect ist auf jeder Hardwareplattform einsetzbar, für die eine Java2-Laufzeitumgebung (JRE) ab Version 1.4.2 zur Verfügung steht.

2.2 Installation

Die Softwarebibliothek CORE.connect wird für alle Plattformen als ZIP-Archiv ausgeliefert. Für die Installation ist das Archiv in ein geeignetes Installationsverzeichnis zu entpacken. Weitere Installationsschritte sind nicht notwendig. Nach der Installation gibt es die folgenden Unterverzeichnisse:

beispiele	enthält XML-Dateien mit Beispieldaten, Beispielprogramme
doc	enthält die Javadoc-Dokumentation für die Bibliothek
jni	enthält die C/C++-Schnittstelle für die Bibliothek
lib	enthält die notwendigen JAR-Bibliotheken

2.3 Deinstallation

Die Softwarebibliothek CORE.connect wird durch Löschen des Installationsverzeichnisses deinstalliert.

3 Funktionsumfang

Die vorliegende Softwarebibliothek CORE.connect ist eine native Java-Bibliothek und für die Verwendung in Java-Software konzipiert. Die zugehörigen JAR-Archive befinden sich im Verzeichnis `lib`. Über die JNI-Schnittstelle von Java wird zusätzlich eine C/C++-Schnittstelle angeboten, über die die Bibliothek auch in anderen Programmierumgebungen (C, C++, Visual Basic, Delphi) genutzt werden kann. Alle notwendigen Dateien einschließlich Beispielen für die C/C++-Schnittstelle sind im Verzeichnis `jni` zu finden. Neben der Java-Variante wird auch eine native .NET-Variante der Softwarebibliothek CORE.connect mit ähnlichem Funktionsumfang angeboten.

Die Softwarebibliothek CORE.connect bietet z.Zt. die folgenden Funktionalitäten:

1. Erzeugung einer Datenlieferung durch Vorgabe eines DatML/RAW-Dokuments oder durch andere XML-Formate nach Transformation mit einem entsprechenden XSLT-Stylesheet.
2. Validierung einer erzeugten Datenlieferung gegen die Dokumenttypdefinition von DatML/RAW. Die Dokumenttypdefinition ist in die Bibliothek integriert.
3. Überprüfung einer erzeugten Datenlieferung gemäß den Vorgaben in einer Liefervereinbarung. Die Liefervereinbarung beschreibt für jede Statistik, welche Merkmale zu melden sind und in welchen Wertebereichen die Merkmalswerte liegen müssen. Durch die Prüfung gegen die Liefervereinbarung wird sichergestellt, dass die versendete Datenlieferung keine groben inhaltlichen Fehler enthält. Die Liefervereinbarungen liegen im XML-Format DatML/SDF vor (s. [SPEZ]) und werden von der Softwarebibliothek bei der Überprüfung automatisch ausgewertet (s. Kap. 6).
4. Versand der erzeugten und geprüften Datenlieferung an den gemeinsamen Dateneingang der Statistik Deutschlands. Die Übertragung erfolgt per HTTPS mit automatischer Authentifizierung beim Empfänger mit separat bereitgestellter Kennung und Passwort. Als Antwort auf die Datenlieferung erhält der Absender einen Eingangsstempel zurück. Mit diesem Schlüssel kann der Absender zu einem späteren Zeitpunkt das Prüfprotokoll für die Datenlieferung beim gemeinsamen Dateneingang abholen.
5. Anforderung eines Prüfprotokolls zu einer Datenlieferung vom gemeinsamen Dateneingang der Statistik Deutschlands. Jede Datenlieferung wird beim Empfänger auf Korrektheit und Vollständigkeit geprüft. Das Ergebnis dieser Überprüfung kann der Absender durch Angabe des zu der Datenlieferung gehörenden Eingangsstempels anfordern. Das Prüfprotokoll wird im XML-Format DatML/RES geliefert.

Weitere Informationen zu der Kommunikationsschnittstelle des zentralen Online-Dateneingangs von eSTATISTIK.core finden sich in der zugehörigen Beschreibung beim Statistik-Portal www.statistik-portal.de.

4 Überblick über die Java-Pakete und -Klassen

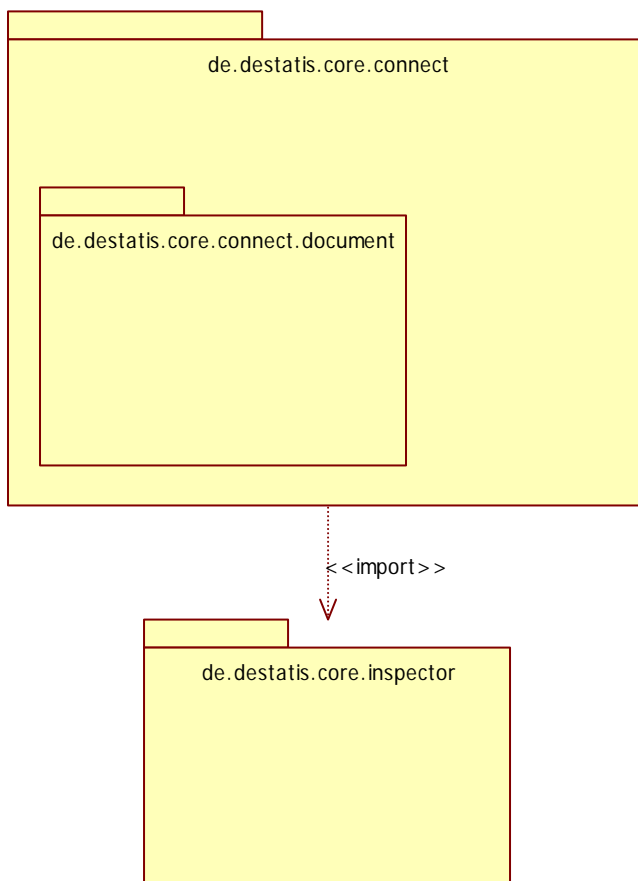
Gemäß dem dargestellten Funktionsumfang gliedert sich die Softwarebibliothek in die folgenden Funktionsbereiche und Java-Pakete:

Funktionsbereich	Paket
Kommunikation	de.destatis.core.connect
Dokumente	de.destatis.core.connect.document
Prüfung (Inspector-Modul)	de.destatis.core.inspector

Für die Realisierung der Softwarebibliothek CORE.connect wurde die folgende Open-Source-Software verwendet:

- HTTPClient 3.0, URL: <http://jakarta.apache.org/commons/httpclient>
- Xerces und Xalan, URL: <http://xml.apache.org>
- Log4j, URL: <http://logging.apache.org>
- Jakarta Commons, URL: <http://jakarta.apache.org/commons>

Paketübersicht:



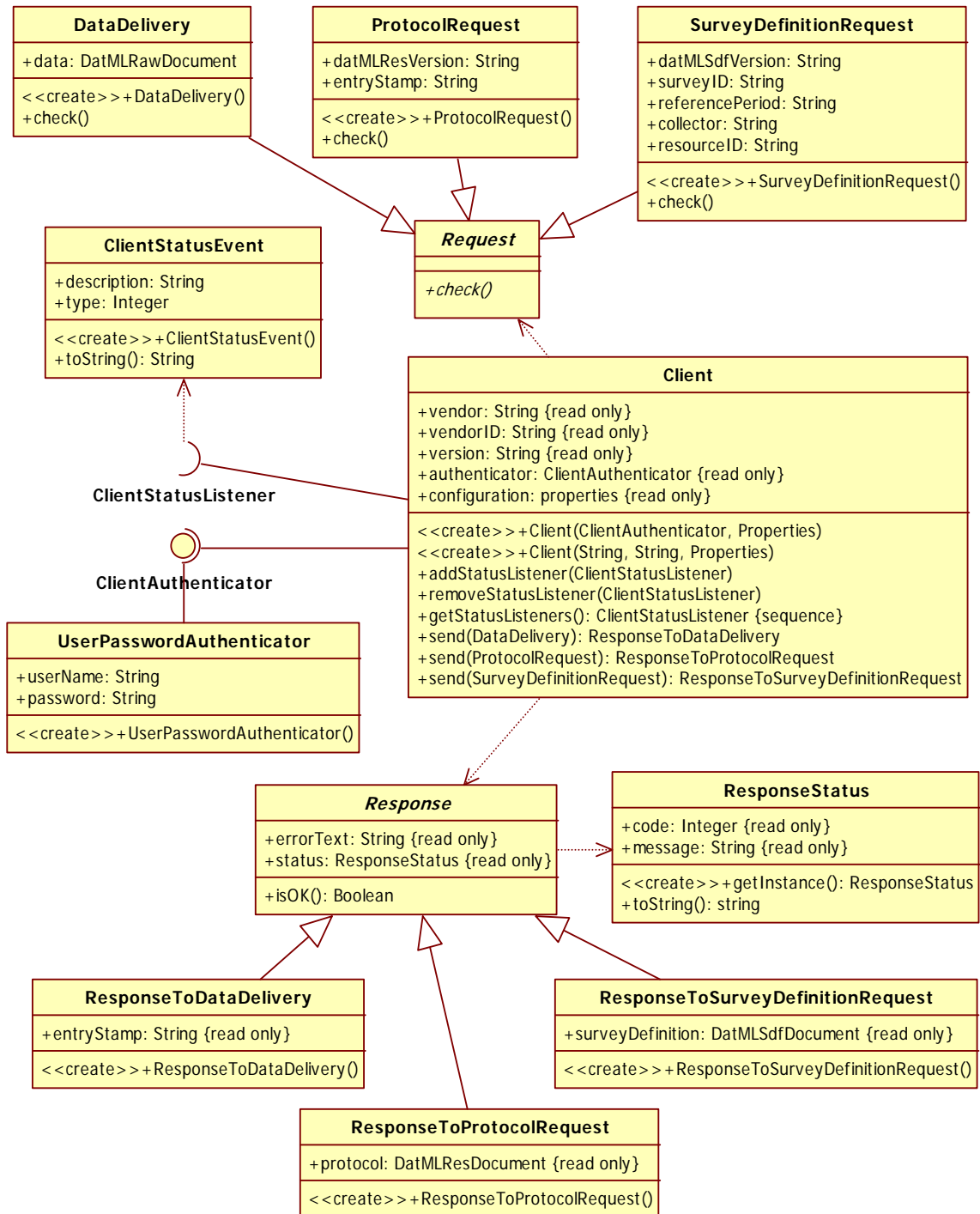
4.1 Paket `de.destatis.core.connect`

Dieses Paket umfasst alle Klassen und Interfaces zur Implementierung der Kommunikationsschnittstelle des zentralen Online-Dateneingangs.

Name der Klasse	Beschreibung
Client	Hauptklasse des Pakets. Implementiert das Kommunikationsprotokoll des CORE-Servers und erlaubt den Versand aller unterstützten Anfragen
DataDelivery	Implementiert eine Anfrage zum Senden einer DatML/RAW-Datenlieferung
ProtocolRequest	Implementiert eine Anfrage zum Anfordern eines DatML/RES-Prüfprotokolls
SurveyDefinitionRequest	Implementiert eine Anfrage zum Anfordern einer DatML/SDF-Erhebungsbeschreibung
ResponseStatus	Stellt Informationen über den Erfolg einer Anfrage (Status) zur Verfügung
ResponseToDataDelivery	Implementiert die Antwort auf die Übermittlung einer DatML/RAW-Datenlieferung
ResponseToProtocolRequest	Implementiert die Antwort auf die Anforderung eines DatML/RES-Prüfprotokolls
ResponseToSurveyDefinitionRequest	Implementiert die Antwort auf die Anforderung einer DatML/SDF-Erhebungsbeschreibung
ClientStatusEvent	Stellt Informationen über den aktuellen Status einer Anfrage zur Verfügung
UserPasswordAuthenticator	Standardimplementierung des Interface ClientAuthenticator

Name des Interface	Beschreibung
ClientAuthenticator	Schnittstelle für die Bereitstellung von Kennung und Passwort für die Authentifizierung beim CORE-Server
ClientStatusListener	Schnittstelle für die Bereitstellung von aktuellen Statusinformationen über die Kommunikation mit dem CORE-Server

Klassendiagramm de.destatis.core.connect:



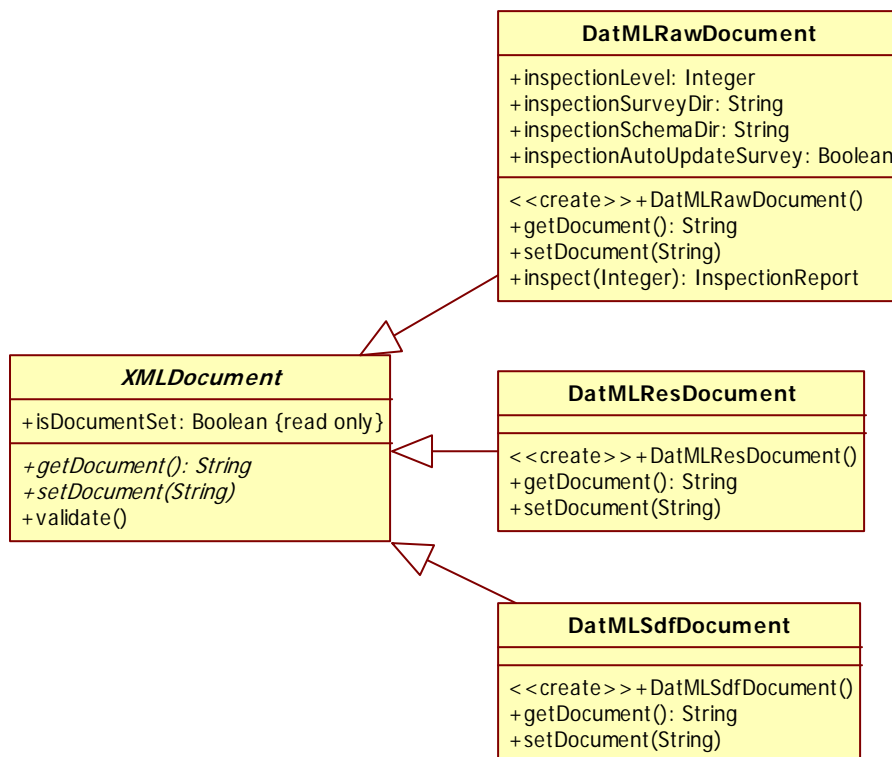
4.2 Paket `de.destatis.core.connect.document`

Dieses Paket umfasst alle Klassen zur Implementierung der verschiedenen verwendeten Dokumententypen DatML/RAW, DatML/RES und DatML/SDF. Das Paket besteht aus den folgenden wichtigsten Klassen.

Name der Klasse	Beschreibung
DatMLRawDocument	Implementiert DatML/RAW-Dokumente
DatMLResDocument	Implementiert DatML/RES-Dokumente
DatMLSdfDocument	Implementiert DatML/SDF-Dokumente

Für die Erzeugung und Auswertung von DatML/RAW- und DatML/RES-Dokumenten werden unter www.statistik-portal.de -> Online-Erhebungen -> CORE -> Download -> XML-Datenformate auf XMLBeans (s. <http://xmlbeans.apache.org>) beruhende Bibliotheken angeboten.

Klassendiagramm `de.destatis.core.connect.document`:



4.3 Paket `de.destatis.core.inspector`

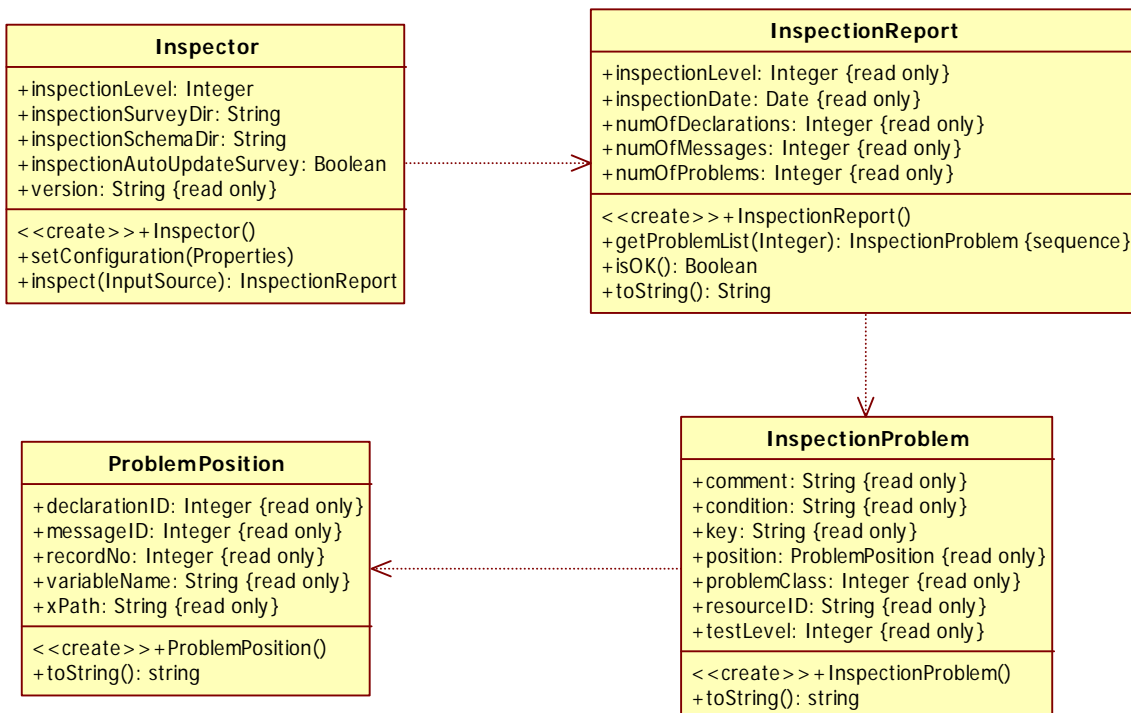
Dieses Paket umfasst alle Klassen zur Implementierung des Inspector-Moduls. Das Inspector-Modul bietet die Möglichkeit, ein DatML/RAW-Dokument hinsichtlich folgender Kriterien zu prüfen:

1. **Syntax:** Wohlgeformtheit und Gültigkeit gemäß XML-Format DatML/RAW.
2. **Semantik:** Einhaltung aller weiteren in der DatML/RAW-Spezifikation festgelegten Formatvorgaben.
3. **Autorisierung:** Einhaltung allgemeiner, nicht statistikspezifischer Vorgaben zur Datenlieferung mit DatML/RAW.
4. **Daten:** Einhaltung der Vorgaben lt. der zur jeweiligen Meldung gehörenden Erhebungsbeschreibung, also z.B. Anzahl und Art der anzugebenden Merkmale und Merkmalsgruppen, Angabe des Berichtszeitraums und der Erhebung.

Eine detaillierte Beschreibung des Inspector-Moduls findet sich in Kap. 7. Das Paket besteht aus den folgenden wichtigsten Klassen.

Name der Klasse	Beschreibung
Inspector	Hauptklasse des Inspector-Moduls, die die Prüffunktionen des Inspector-Moduls implementiert
InspectionReport	Vom Inspector-Modul erstellte Zusammenfassung der Prüfergebnisse eines DatML/RAW-Dokuments
InspectionProblem	Beschreibt ein bei der Prüfung festgestelltes Problem
ProblemPosition	Beschreibt die Position eines festgestellten Problems in dem geprüften DatML/RAW-Dokument

Klassendiagramm `de.destatis.core.inspector`:



5 Konfigurationseinstellungen

Die Funktionen der Softwarebibliothek CORE.connect werden über Konfigurationseinstellungen gesteuert. Diese Einstellungen können, sofern sie von den Standardeinstellungen abweichen, bei der Instanziierung der beiden Klassen `Client` und `Inspector` in Form von `Properties`-Objekten mitgegeben sowie teilweise über entsprechende Methoden individuell eingestellt werden.

Im Folgenden werden alle Konfigurationseinstellungen mit ihrem Schlüssel, den möglichen Schlüsselwerten sowie ihre Bedeutung beschrieben.

Schlüssel: `client.username`
Java-Konstante: `PROPERTY_USERNAME`
Wertebereich: Beliebige Zeichenkette
Standardwert: -
Bedeutung: Mit diesem Schlüssel wird die Benutzerkennung eingestellt, die bei der Kommunikation mit dem zentralen Online-Dateneingang verwendet werden soll.

Schlüssel: `client.password`
Java-Konstante: `PROPERTY_PASSWORD`
Wertebereich: Beliebige Zeichenkette
Standardwert: -
Bedeutung: Mit diesem Schlüssel wird das Benutzerpasswort eingestellt, das bei der Kommunikation mit dem zentralen Online-Dateneingang verwendet werden soll.

Schlüssel: `client.compression`
Java-Konstante: `PROPERTY_COMPRESSION`
Wertebereich: `none`, `deflate`, `gzip`
Standardwert: `none`
Bedeutung: Mit diesem Schlüssel kann eingestellt werden, ob die Daten komprimiert übertragen werden sollen.

Schlüssel: `client.address.useURI`
Java-Konstante: `PROPERTY_ADDRESS_USE_URI`
Wertebereich: `false`, `true`
Standardwert: `false`
Bedeutung: Mit diesem Schlüssel kann eingestellt werden, ob für den Verbindungsaufbau zum zentralen Online-Dateneingang die IP-Adresse oder der DNS-Name des CORE-Servers verwendet werden soll.

Schlüssel: `client.protocol`
Java-Konstante: `PROPERTY_PROTOCOL`
Wertebereich: `https`, `osci`
Standardwert: `https`
Bedeutung: Mit diesem Schlüssel kann eingestellt werden, welches Protokoll für die Datenübertragung an den zentralen Online-Dateneingang verwendet werden soll. Das Übertragungsprotokoll `osci` wird z.Zt. noch nicht unterstützt.

Schlüssel: `client.certificate.dir`
Java-Konstante: `PROPERTY_CERTIFICATE_DIR`
Wertebereich: Gültiger Verzeichnispfad
Standardwert: Internes Root-Zertifikat verwenden
Bedeutung: Mit diesem Schlüssel kann eingestellt werden, in welchem Verzeichnis weitere Root-Zertifikate für die Überprüfung des Server-Zertifikats des zentralen Online-Dateneingangs gesucht werden sollen. Diese Einstellung ist eine systemweite Einstellung, die beim Start der JVM als System-Property gesetzt und danach nicht mehr geändert werden kann.

Schlüssel: `client.log.file`
Java-Konstante: `PROPERTY_LOG_FILE`
Wertebereich: Gültiger Dateiname oder `CONSOLE`
Standardwert: -
Bedeutung: Mit diesem Schlüssel kann der Name einer Datei angegeben werden, in die alle Protokollmeldungen gemäß der eingestellten Protokollstufe gespeichert werden. Bei Angabe von `CONSOLE` werden die Meldungen auf der Konsole ausgegeben. Wenn keine Protokolldatei angegeben ist, erfolgt auch keine Protokollierung. Diese Einstellung ist eine systemweite Einstellung, die beim Start der JVM als System-Property gesetzt und danach nicht mehr geändert werden kann.

Schlüssel: `client.log.level`
Java-Konstante: `PROPERTY_LOG_LEVEL`
Wertebereich: 0, 1, 2
Standardwert: 0
Bedeutung: Mit diesem Schlüssel kann der Detaillierungsgrad der Protokollierung eingestellt werden. Je höher der Detaillierungsgrad, desto detaillierter sind die Protokollmeldungen. Bei der Stufe 2 (=Debug) werden auch die zwischen Client und Server ausgetauschten Daten protokolliert. Wenn keine Protokolldatei angegeben ist, erfolgt auch keine Protokollierung. Diese Einstellung ist eine systemweite Einstellung, die beim Start der JVM als System-Property gesetzt und danach nicht mehr geändert werden kann.

Die folgenden Konfigurationseinstellungen sind nur relevant, wenn die Kommunikation mit dem zentralen Online-Dateneingang über einen Proxy-Server erfolgt.

Schlüssel: `client.proxy.user`
Java-Konstante: `PROPERTY_PROXY_USER`
Wertebereich: Beliebige Zeichenkette
Standardwert: -
Bedeutung: Mit diesem Schlüssel wird die Benutzerkennung eingestellt, die für die Authentifizierung bei einem Proxy-Server verwendet werden soll.

Schlüssel: `client.proxy.password`
Java-Konstante: `PROPERTY_PROXY_PASSWORD`
Wertebereich: Beliebige Zeichenkette
Standardwert: -

- Bedeutung:** Mit diesem Schlüssel wird das Benutzerpasswort eingestellt, das für die Authentifizierung bei einem Proxy-Server verwendet werden soll.
- Schlüssel:** `client.proxy.host`
- Java-Konstante:** `PROPERTY_PROXY_HOST`
- Wertebereich:** Gültiger DNS-Name oder IP-Adresse
- Standardwert:** -
- Bedeutung:** Mit diesem Schlüssel wird der Netzwerkname des Proxy-Servers eingestellt, über den die Kommunikation mit dem zentralen Online-Dateneingang erfolgen soll.
- Schlüssel:** `client.proxy.port`
- Java-Konstante:** `PROPERTY_PROXY_PORT`
- Wertebereich:** Gültige Portnummer
- Standardwert:** 80
- Bedeutung:** Mit diesem Schlüssel wird der Port des Proxy-Servers eingestellt, über den die Kommunikation mit dem zentralen Online-Dateneingang erfolgen soll.
- Schlüssel:** `client.proxy.authmethod`
- Java-Konstante:** `PROPERTY_PROXY_AUTH_METHOD`
- Wertebereich:** auto, basic, digest, ntlm
- Standardwert:** auto
- Bedeutung:** Mit diesem Schlüssel wird eingestellt, welche Authentifizierungsmethode beim Proxy-Server bei der Kommunikation mit dem zentralen Dateneingang zu verwenden ist. Bei der Methode `NTLM` wird nur die Version 1 unterstützt. Bei Angabe des Wertes `auto` wird die Authentifizierungsmethode automatisch bestimmt, indem eine Authentifizierung in der Reihenfolge `ntlm`, `digest` und `basic` versucht wird.
- Schlüssel:** `client.proxy.domain`
- Java-Konstante:** `PROPERTY_PROXY_DOMAIN`
- Wertebereich:** Gültige Domainname
- Standardwert:** -
- Bedeutung:** Mit diesem Schlüssel wird der bei der Authentifizierungsmethode `NTLM` benötigte Domainname eingestellt. Bei allen anderen Authentifizierungsmethoden wird dieser Schlüssel ignoriert.

Die folgenden Konfigurationseinstellungen steuern das Verhalten des Inspector-Moduls, mit dem weitreichende Prüfungen von DatML/RAW-Dokumenten vor dem Versand durchgeführt werden können.

- Schlüssel:** `inspector.inspectionLevel`
- Java-Konstante:** `PROPERTY_INSPECTION_LEVEL`
- Wertebereich:** 1, 2, 3, 4
- Standardwert:** 2
- Bedeutung:** Mit diesem Schlüssel wird die Prüfstufe bei der Inspektion von DatML/RAW-Daten eingestellt. Bei Stufe 4 werden die Daten am strengsten geprüft

Schlüssel: `inspector.survey.dir`
Java-Konstante: `PROPERTY_SURVEY_DIR`
Wertebereich: Gültiger Verzeichnispfad
Standardwert: -
Bedeutung: Mit diesem Schlüssel wird das Verzeichnis eingestellt, in dem bei der Inspektion von DatML/RAW-Daten nach Erhebungsbeschreibungen gesucht wird. Diese Einstellung ist nur relevant, wenn die Prüfstufe 4 verwendet wird.

Schlüssel: `inspector.schema.dir`
Java-Konstante: `PROPERTY_SCHEMA_DIR`
Wertebereich: Gültiger Verzeichnispfad
Standardwert: Interne Schemadateien
Bedeutung: Mit diesem Schlüssel wird das Verzeichnis eingestellt, in dem bei der Validierung von XML-Daten nach den notwendigen XML-Schemadateien gesucht wird. Im Normalfall sind die intern bereitgestellten Schemadateien hinreichend.

Schlüssel: `inspector.survey.autoupdate`
Java-Konstante: `PROPERTY_SURVEY_AUTOUPDATE`
Wertebereich: no, yes
Standardwert: no
Bedeutung: Mit diesem Schlüssel wird eingestellt, ob bei der Inspektion von DatML/RAW-Daten automatisch nach aktuelleren Erhebungsbeschreibungen auf dem CORE-Server gesucht werden soll. Diese Einstellung wird z.Zt. noch nicht unterstützt.

6 Statuscodes des zentralen Online-Dateneingangs

Bei der Kommunikation mit dem zentralen Online-Dateneingang enthält das durch die `Client`-Methode `getStatus` gelieferte `ResponseStatus`-Objekt einen anwendungsspezifischen Statuscode, der Auskunft über den Erfolg der jeweiligen Anfrage gibt. Dieser Statuscode kann die folgenden Werte und Bedeutungen haben (s. [KOMM] oder Java-Doc von `ResponseStatus`).

Statuscode (Wert, Java-Konstante)	Bedeutung
0 = OK	Es ist kein Fehler aufgetreten. Die Verarbeitung der Anfrage beim gemeinsamen Dateneingang war erfolgreich.
10 = BAD_REQUEST	Die Art und Anzahl der übermittelten <code>multipart/form-data</code> -Teile und/oder Feldnamen waren nicht korrekt. Es wurden z.B. Felder vergessen oder falsch angegeben.
20 = LOGIN_ERROR	Bei der Authentifizierung beim gemeinsamen Dateneingang ist ein Fehler aufgetreten, z.B. weil Kennung und Passwort falsch angegeben wurden.
30 = BAD_RESPONSE	Die vom gemeinsamen Dateneingang empfangene Antwort ist fehlerhaft und kann nicht korrekt weiterverarbeitet werden.
100 = FILE_RECEIVE_ERROR	Die übermittelten Daten konnten nicht verarbeitet werden, z.B. weil der angegebene Zeichensatz unbekannt ist oder die Daten zu umfangreich.
110 = NO_VALID_XML	Die übermittelten Daten sind kein gültiges DatML/RAW-Dokument.
200 = RES_NOT_AVAILABLE	Für die angegebene Datenlieferung ist noch kein Prüfprotokoll vorhanden, weil es noch nicht erstellt ist. Der Abruf des Prüfprotokolls sollte später wiederholt werden.
210 = RES_FORMAT_ERROR	Die angegebene Version des Formats DatML/RES wird nicht (mehr) unterstützt.
220 = RES_INVALID_ID	Dem gesendeten Eingangsstempel ist keine Datenlieferung des Absenders zugeordnet, wodurch auch kein Prüfprotokoll bereitgestellt werden kann.
230 = RES_DELETED	Für die angegebene Datenlieferung war ein Prüfprotokoll vorhanden, das aber inzwischen wegen Ablauf der Aufbewahrungsfrist gelöscht wurde.
300 = SDF_NOT_AVAILABLE	Für die angegebene Erhebung und Berichtszeitraum ist keine Erhebungbeschreibung vorhanden, z.B. weil eine ungültige Erhebung oder Berichtszeitraum angegeben wurde.
310 = SDF_FORMAT_ERROR	Die angegebene Version des Formats DatML/SDF wird nicht (mehr) unterstützt.
320 = SDF_UPTODATE	Für die angegebene Erhebung und Berichtszeitraum liegt keine aktuellere Erhebungsbeschreibung vor.

Wenn ein Fehler auf der Ebene des Übertragungsprotokolls, also HTTP(S), auftritt, dann werden der HTTP-Statuscode und -Fehlertext in der dabei ausgelösten Ausnahme `TransferProtocolException` mitgeteilt. Ein `ResponseStatus`-Objekt steht in einem solchen Fall nicht zur Verfügung.

7 Fehlerschlüssel und -meldungen des Inspector-Moduls

Das Inspector-Modul (Klasse `Inspector`) ist eine konfigurierbare Komponente der Softwarebibliothek CORE.connect, mit der DatML/RAW-Daten vor dem Versand einer eingehenden Prüfung unterzogen werden können. Diese Prüfungen beruhen nicht nur auf den Angaben in der XML-Schemadatei von DatML/RAW, sondern können auch die zu jeder Statistik bereitgestellten Erhebungsbeschreibungen nutzen.

Das Inspector-Modul untersucht DatML/RAW-Daten gemäß vorgegebener Prüfstufen und liefert im Ergebnis einen Prüfbericht (Klasse `InspectionReport`) mit einer Liste von Prüfergebnissen (Klasse `InspectionProblem`). Die Prüfung von DatML/RAW-Daten erfolgt auf den folgenden Prüfebene:

5. **Syntax**: Wohlgeformtheit und Gültigkeit gemäß XML-Format DatML/RAW.
6. **Semantik**: Einhaltung aller weiteren in der DatML/RAW-Spezifikation festgelegten Formatvorgaben.
7. **Autorisierung**: Einhaltung allgemeiner, nicht statistikspezifischer Vorgaben zur Datenlieferung mit DatML/RAW.
8. **Daten**: Einhaltung der Vorgaben lt. der zur jeweiligen Meldung gehörenden Erhebungsbeschreibung, also z.B. Anzahl und Art der anzugebenden Merkmale und Merkmalsgruppen, Angabe des Berichtszeitraums und der Erhebung.

Aufbauend auf diesen Prüfebene definieren sich die vier Prüfstufen des Moduls:

Prüfstufe	Prüfebene
1	Syntax (1)
2	Syntax (1) + Semantik (2)
3	Syntax (1) + Semantik (2) + Autorisierung (3)
4	Syntax (1) + Semantik (2) + Autorisierung (3) + Daten (4)

Das Modul erhält das zu untersuchende DatML/RAW-Dokument und die zu verwendende Prüfstufe und liefert darauf hin einen entsprechenden Prüfbericht. Der Prüfbericht besteht aus summarischen Angaben zur Prüfung sowie einer Liste mit detaillierten Prüfergebnissen. Die Prüfergebnisse sind in vier Problemklassen unterteilt:

Problemklasse	Code=Kürzel	Bedeutung
Information	1=INFO	Information oder Hinweis
Warnung	2=WARN	Warnung
Fehler	3=ERROR	Fehler
Abbruch	4=FATAL	Nicht behebbarer Fehler

Neben der Problemklasse enthält ein Prüfergebnis die folgenden Informationen:

1. die Prüfebene, auf der das Problem festgestellt wurde,
2. einen eindeutigen Schlüssel zur Identifikation des Problems,
3. einer textuelle Erläuterung des Problems, sowie
4. die Position im Dokument, an der das Problem festgestellt wurde.

Die Position (Klasse `ProblemPosition`) im Dokument, an der ein Problem festgestellt wurde, wird in logischer Form durch eine Nachrichten-, Meldungs- und Satznummer sowie einen absoluten XPath-Ausdruck angegeben. Wenn sich ein Problem keiner Nachricht oder Meldung oder Satz zuordnen lässt, dann wird als Nummer 0 verwendet.

Im Folgenden werden die einzelnen Prüfebene und die möglichen Prüfergebnisse mit Problemkategorie, Problemschlüssel und Fehlertext erläutert.

7.1 Prüfebene: Syntax

Auf dieser Prüfebene wird das DatML/Raw-Dokument auf Wohlgeformtheit und Gültigkeit gemäß dem zugehörigen XML-Schema geprüft.

Für die Ergebnisse der Überprüfung werden die folgenden Schlüssel und Texte verwendet:

Kategorie	Schlüssel	Text
WARN	12001	Hier wird der Originaltext der <code>SAXParseException</code> übernommen. Alle Warnungen haben nur einen Schlüssel.
ERROR	13001	Hier wird der Originaltext der <code>SAXParseException</code> übernommen. Alle Fehler haben nur einen Schlüssel.
FATAL	14001	Hier wird der Originaltext der <code>SAXParseException</code> übernommen. Alle nicht behebbaren Fehler haben nur einen Schlüssel.

Die verschiedenen Arten von Parserfehlern werden in den Schlüsseln nicht weiter ausdifferenziert. Alle Parserfehler einer Kategorie erhalten einen einheitlichen Schlüssel. Zur Identifikation der Position des Prüfergebnisses werden die Nachrichten-, Meldungs- und Satznummer ausgewiesen sowie der absolute XPath-Ausdruck. Die Zeilen- und Spaltennummer aus der `SAXParseException` wird in den Text übernommen.

7.2 Prüfebene: Semantik

Auf dieser Prüfebene wird das DatML/Raw-Dokument auf die Einhaltung aller weiteren, nicht im XML-Schema beschriebenen, Anforderungen geprüft (s. [RAW]). Das sind:

Für die Ergebnisse der Überprüfung werden die folgenden Schlüssel und Texte verwendet:

Kategorie	Schlüssel	Text
WARN	22001	Die Anzahl der Nachrichten muss mit der Angabe im Element <code>anzahl</code> übereinstimmen.
ERROR	23001	Jeder Meldung muss genau ein Element <code>erhebung</code> zugeordnet sein.
ERROR	23002	Jeder Meldung muss genau ein Element <code>berichtszeitraum</code> zugeordnet sein.
ERROR	23003	Jeder Meldung darf höchstens ein Element <code>berichtspflichtiger</code> zugeordnet sein.
ERROR	23004	Jeder Meldung darf höchstens ein Element <code>berichtsempfaenger</code> zugeordnet sein.
ERROR	23005	Jeder Meldung darf höchstens ein Element <code>material</code> zugeordnet sein.
ERROR	23006	Jeder Meldung darf höchstens ein Element <code>datenattribute</code> zugeordnet sein.
ERROR	23007	Ein Hilfsmerkmal muss einen Namen (Attribut <code>name</code>) haben.
ERROR	23008	Die Hilfsmerkmale einer Meldung müssen verschiedene Namen (Attribut <code>name</code>) haben.
ERROR	23009	Ein Merkmal muss einen Namen (Attribut <code>name</code>) haben.
ERROR	23010	Ein Merkmal (Attribut <code>name</code>) darf nur einmal in einem Datensatz oder einer Merkmalsgruppe vorkommen.
ERROR	23011	Eine Merkmalsgruppe muss einen Namen (Attribut <code>name</code>) haben.
ERROR	23012	Der Index einer Merkmalsgruppe (Attribut <code>index</code>) darf nur numerisch sein oder ein indexbildendes Merkmal in der Form <code>name(<merkmalsname>)</code> .

Kategorie	Schlüssel	Text
ERROR	23013	Das indexbildende Merkmal muss in der Merkmalsgruppe vorkommen und darf nicht leer sein.
ERROR	23014	Der Wert des indexbildenden Merkmals muss numerisch sein.
ERROR	23015	Die Indizes einer Merkmalsgruppe müssen alle verschieden sein.
ERROR	23016	Die Indizierungsmethode muss für alle Instanzen einer Merkmalsgruppe gleich sein.

Zur Identifikation der Position des Prüfergebnisses werden die Nachrichten-, Meldungs- und Satznummer ausgewiesen sowie der absolute XPath-Ausdruck. Bei den Prüfergebnissen mit den Schlüsseln 23007 bis 23016 wird zusätzlich der Name des betroffenen Hilfsmerkmals, Merkmals bzw. Merkmalsgruppe angegeben.

7.3 Prüfebene: Autorisierung

Auf dieser Prüfebene wird das DatML/RAW-Dokument auf die Einhaltung allgemeiner, nicht statistikspezifischer, Vorgaben für DatML/RAW-Datenlieferungen geprüft.

Für die Ergebnisse der Überprüfung werden die folgenden Schlüssel und Texte verwendet:

Kategorie	Schlüssel	Text
ERROR	33001	Die Absenderkennung muss mit der IDEV/CORE-Kennung übereinstimmen.
ERROR	33002	Für die Testkennung sind nur die Werte 100 oder 200 zulässig.
ERROR	33003	Die Erhebungskennung muss einen der Werte 0001, ..., 9999 enthalten.
ERROR	33004	Die Klasse der Erhebungskennung muss den Wert ERHID haben.
ERROR	33005	Die Absenderkennung darf nicht leer sein.
ERROR	33006	Die Klasse der Absender- und Berichtspflichtigenkennung muss den Wert MELDID haben.
ERROR	33007	Die Empfängererkennung muss angegeben werden.
ERROR	33008	Die Empfängererkennung muss einen der Werte 00,..., 16, 99 enthalten.
ERROR	33009	Bei Datenlieferungen an den gemeinsamen Dateneingang muss zu jeder Meldung ein Berichtsempfänger angegeben werden.
ERROR	33010	Wenn ein Berichtsempfänger angegeben ist, dann muss auch die Kennung angegeben werden.
ERROR	33011	Die Berichtsempfängererkennung muss einen der Werte 00,..., 16 enthalten.
ERROR	33012	Die Klasse der Empfänger- und Berichtsempfängererkennung muss den Wert STAID haben.
ERROR	33013	Der Name der Organisation oder Person muss angegeben werden.
ERROR	33014	Die Postleitzahl und der Ort müssen angegeben werden.
ERROR	33015	Der Berichtszeitraum ist der Zeitraum, für den berichtet wird, und muss daher mindestens begonnen haben.

Zur Identifikation der Position des Prüfergebnisses werden die Nachrichten- und Meldungsnummer ausgewiesen sowie der absolute XPath-Ausdruck.

7.4 Prüfebene: Daten

Auf dieser Prüfebene wird das DatML/RAW-Dokument auf Einhaltung der Vorgaben gemäß der zu jeder Meldung gehörenden Erhebungsbeschreibung geprüft.

Für die Ergebnisse der Überprüfung werden die folgenden Schlüssel und Texte verwendet:

Kategorie	Schlüssel	Text
WARN	42001	Es sind keine Erhebungsbeschreibungen vorhanden.
WARN	42002	Zu der Erhebung <eeee> sind keine Erhebungsbeschreibungen vorhanden.
WARN	42003	Für die Erhebung <eeee> und den Berichtszeitraum <Jahr, Halbjahr, Semester, Quartal, Monat, Woche> gibt es keine Erhebungsbeschreibung.
WARN	42004	Für die Erhebung <eeee> und den angegebenen Berichtszeitraum gibt es keine Erhebungsbeschreibung.
WARN	42005	Für die Erhebung <eeee> und den Berichtsempfänger <bb> gibt es keine Erhebungsbeschreibung.
ERROR	43001	Das Format der Erhebungsbeschreibung ist kein gültiges DatML/SDF.
ERROR	43002	Dateiname und Inhalt der Erhebungsbeschreibung passen nicht zusammen.
ERROR	43003	Die verwendete Notationssprache für Bedingungen wird nicht unterstützt.
ERROR	43004	Der Wert des Hilfsmerkmals liegt nicht im definierten Wertebereich.
ERROR	43005	Das Hilfsmerkmal muss angegeben werden.
ERROR	43006	Die Bedingung für die Angabe des Hilfsmerkmals ist erfüllt, so dass das Hilfsmerkmal angegeben werden muss.
ERROR	43007	Die Bedingung für die Angabe des Hilfsmerkmals ist nicht erfüllt und strikt anzuwenden, so dass das Hilfsmerkmal nicht angegeben werden darf.
ERROR	43008	Das Hilfsmerkmal ist an dieser Stelle unzulässig.
ERROR	43009	Der Satz muss einer der definierten Satzarten entsprechen.
ERROR	43010	Der Wert des Merkmals liegt nicht im definierten Wertebereich.
ERROR	43011	Das Merkmal muss angegeben werden.
ERROR	43012	Die Bedingung für die Angabe des Merkmals ist erfüllt, so dass das Merkmal angegeben werden muss.
ERROR	43013	Die Bedingung für die Angabe des Merkmals ist nicht erfüllt und strikt anzuwenden, so dass das Merkmal nicht angegeben werden darf.
ERROR	43014	Das Merkmal ist an dieser Stelle unzulässig.
ERROR	43015	Die Merkmalsgruppe muss <direkt, indirekt, durch Name> indiziert werden.
ERROR	43016	Das angegebene indexbildende Merkmal ist für die Merkmalsgruppe unzulässig.
ERROR	43017	Von der Merkmalsgruppe müssen/dürfen mindestens/genau/höchstens <n> Instanzen angegeben werden.
ERROR	43018	Die Merkmalsgruppe muss angegeben werden.
ERROR	43019	Die Bedingung für die Angabe der Merkmalsgruppe ist erfüllt, so dass die Merkmalsgruppe angegeben werden muss.
ERROR	43020	Die Bedingung für die Angabe der Merkmalsgruppe ist nicht erfüllt und strikt anzuwenden, so dass die Merkmalsgruppe nicht angegeben werden darf.
ERROR	43021	Die Merkmalsgruppe ist an dieser Stelle unzulässig.
ERROR	43022	Es müssen/dürfen mindestens/genau/höchstens <n> Datensätze geliefert werden.

Kategorie	Schlüssel	Text
FATAL	44001	Das konfigurierte Verzeichnis mit den verfügbaren Erhebungsbeschreibungen ist ungültig oder nicht vorhanden.

Zur Identifikation der Position des Prüfergebnisses werden die Nachrichten-, Meldungs- und Satznummer ausgewiesen sowie der absolute XPath-Ausdruck. Bei den Prüfergebnissen mit den Schlüsseln 43004 bis 43018 wird zusätzlich der Name des betroffenen Hilfsmerkmals, Merkmals bzw. Merkmalsgruppe angegeben.

8 Weitere Dokumentation

Für die Softwarebibliothek CORE.connect befindet sich im Verzeichnis `doc` die zugehörige Javadoc-Dokumentation.

8.1 Beispiele

Im Verzeichnis `beispiele` finden sich DatML/RAW-Dateien mit Beispieldaten für die Vierteljährliche Verdiensterhebung (`vve*.xml`), den Monatsbericht im Verarbeitenden Gewerbe (`mbb*.xml`) und die Monatserhebung im Einzelhandel (`mhg-eh*.xml`). Die Anforderungen an den Inhalt von Lieferungen für diese Statistiken werden in der zugehörigen Erhebungsbeschreibungen (`0001*.xml`, `0003*.xml` und `0018*.xml`) beschrieben.

Im Unterverzeichnis `src/test` befinden sich zwei Java-Beispielprogramme, die die Benutzung der Softwarebibliothek demonstrieren. Die Konfigurationsdatei `client.ini` wird von den Beispielprogrammen zur Konfiguration verwendet. Die Batch-Datei `client.bat` zeigt, wie der Versand von Datenlieferungen und der Abruf von Prüfprotokollen ohne zusätzliche Programmierung direkt von der Kommandozeile aus durchgeführt werden kann. Mit dem Batch-Programm `inspector.bat` können beliebige Datenlieferungen (DatML/RAW-Dokumente) auf ihre Richtigkeit bzgl. der zugehörigen Erhebungsbeschreibung überprüft werden. Weitere Erhebungsbeschreibungen im DatML/SDF-Format sowie die zugehörigen Liefervereinbarungen im PDF-Format können vom Statistik-Portal unter www.statistik-portal.de -> Online-Erhebungen -> CORE -> Liefervereinbarungen heruntergeladen werden.

Für den Versand von Datenlieferungen an den zentralen Online-Dateneingang der Statistik Deutschlands werden eine Kennung und ein Passwort benötigt. Beides kann über das Statistik-Portal www.statistik-portal.de -> Online-Erhebungen -> CORE -> Registrierung angefordert werden.

Der Eingang einer Datenlieferung beim zentralen Online-Dateneingang der statistischen Ämter kann vom Absender folgendermaßen überprüft werden:

1. mit einem Browser die Webadresse <https://www-idev.destatis.de/idev> aufrufen und durch Eingabe von Kennung und Passwort anmelden. Kennung und Passwort stimmen mit denen überein, die auch für den Versand der Datenlieferung verwendet wurden.
2. im Menü 'Benutzerdaten' am rechten oberen Bildschirmrand den Eintrag 'Melderkonto' wählen. Daraufhin wird das Melderkonto angezeigt, in dem alle eingegangenen Meldungen sortiert nach Eingangszeitpunkt verzeichnet sind. Datenlieferungen, die über den gemeinsamen Dateneingang eingegangen sind, enthalten die Angabe `connect-datml.xml` als Name der Originaldatei.

8.2 Weitere Informationen

Weitere Informationen zu den DatML-Datenformaten, zu Liefervereinbarungen für ausgewählte Statistiken und Software sind über das Statistik-Portal www.statistik-portal.de abrufbar.