**XSD Profiling Tool**

**Technische Dokumentation**

**ARS Computer und Consulting GmbH**

**Version: 1.0, 05.01.2011**

**Änderungsprotokoll**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Autor[en]** | **Datum** | **Beschreibung** |

|  |  |  |  |
| --- | --- | --- | --- |
| 1.0 | Martin Beier | 05.01.2011 | Dokumentation erstellt:  1. Projektbeschreibung  2. Implementierung  3. Konfiguration und Anpassung |

**Inhaltsverzeichnis**

[1. Projektbeschreibung 4](#_Toc281920550)

[1.1. eXTra Standard 4](#_Toc281920551)

[1.2. Profilierung 4](#_Toc281920552)

[1.3. XSD Profiling Tool 4](#_Toc281920553)

[2. Implementierung 4](#_Toc281920554)

[1.4. Verwendete Frameworks 4](#_Toc281920555)

[1.1.1. log4j 4](#_Toc281920556)

[1.1.2. Xml Schema Object Model (XSOM) 4](#_Toc281920557)

[1.1.3. iText 4](#_Toc281920558)

[1.5. Packages 4](#_Toc281920559)

[1.1.4. de.drv.dsrv.xtt.gui 4](#_Toc281920560)

[1.1.5. de.drv.dsrv.xtt.gui.model 4](#_Toc281920561)

[1.1.6. de.drv.dsrv.xtt.tester 4](#_Toc281920562)

[1.1.7. de.drv.dsrv.xtt.util 4](#_Toc281920563)

[1.1.8. de.drv.dsrv.xtt.util.pdf 4](#_Toc281920564)

[1.1.9. de.drv.dsrv.xtt.util.schema 4](#_Toc281920565)

[1.1.10. de.drv.dsrv.xtt.gui.util.tools 4](#_Toc281920566)

[3. Konfiguration und Anpassung 4](#_Toc281920567)

[1.6. Anwendungseinstellungen 4](#_Toc281920568)

[1.7. Anmerkungen 4](#_Toc281920569)

# Projektbeschreibung

## eXTra Standard

eXTra, das Einheitliche XML-basierte Transportverfahren, ist ein offener Standard für die Datenübermittlung. Sämtliche öffentlichen Informationen über eXTra sind im Internet unter der Adresse http://www.extra-standard.de abrufbar.

Das Verfahren eXTra definiert ein allgemein gültiges Format einer Beschreibungsstruktur zum Zwecke des Austauschs von Fachnachrichten zwischen zwei Kommunikationspartnern und deren Prozessbeteiligten.

Das sogenannte eXTra Basis-Verfahren entspricht zugleich einer Zusammenfassung wie auch Verallgemeinerung der bestehenden Verfahren. Es obliegt dem jeweiligen Datenübermittlungsverbund, bzw. Fachverfahren, ausgehend vom eXTra Basis-Verfahren, eine auf dessen Belange zugeschnittene Ausprägung – sein spezifi-sches eXTra-Verfahren - zu definieren. Dieser Vorgang des Zuschnitts heißt Profilierung des eXTra Basis-Standards, das Ergebnis einer Profilierung ist ein verbund- bzw. fachspezifischer eXTra Standard.

## Profilierung

Die formale Syntaxbeschreibung des eXTra Basis-Standards ist in XSD-Schemadateien hinterlegt. Im Zuge der Profilierung des eXTra Basis-Standards hin zu einem verbund- bzw. fachspezifischen eXTra Standard werden deshalb die Schemadateien des eXTra Basis-Standards profiliert, d.h. im wesentlichen eingeschränkt und genauer spezifiziert.

Ausgangspunkt ist die jeweils gültige Beschreibung des eXTra-Verfahrens. Hierzu gehören Schema-Dateien des eXTra Basis-Standards, die die Syntax einer gültigen eXTra-Nachricht definieren. Eine Profilierung dieser Schema-Dateien bedeutet, sie so zu modifizieren, dass eine damit geprüfte XML-Datei sowohl dem eXTra-Basis-Standard, als auch den Regeln des so erzeugten verbund- bzw. fachspezifischen eXTra Standards genügt.

## XSD Profiling Tool

Das XSD Profiling Tool ist eine Java-Applikation für die Durchführung einer solchen Profilierung (siehe Abbildung 1: Oberfläche XSD Profiling Tool). Mit Hilfe eines Baumes können die Elemente des eXTra Basis-Standards (request oder response) eingeschränkt und genauer spezifiziert werden.

Das Ergebnis dieser Profilierung wird in einer sogenannten Profilkonfiguration (XML) gespeichert und kann später weiter bearbeitet werden.

Aus einer Profilkonfiguration kann das fachspezifische eXTra Schema generiert werden. Optional erzeugt das XSD Profiling Tool eine PDF-Dokumentation mit den verwendeten Elementen und ggf. im Zuge der Profilierung hinterlegten Anmerkungen.

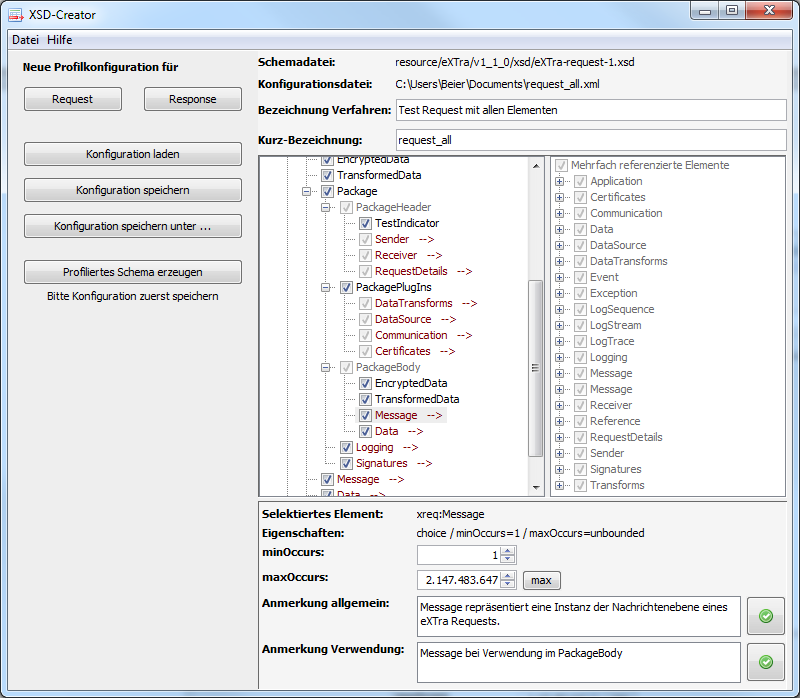


Abbildung 1: Oberfläche XSD Profiling Tool

# Implementierung

XSD Profiling Tool ist ein Java 1.6 Projekt. Neben dem entsprechenden JDK wurden die im folgenden Punkt beschriebenen Frameworks verwendet. Danach sind die Packages des Projekts beschrieben.

## Verwendete Frameworks

### log4j

Für das Logging innerhalb der Anwendung wird das log4j-Framework in der Version 1.2.15 verwendet.

Quelle: <http://logging.apache.org/log4j/>

### Xml Schema Object Model (XSOM)

Zum Parsen der XML-Schemadateien wird die XSOM Java-Bibliothek (Version 20060901) verwendet. Damit werden auch XML-Schemata mit mehreren Namespaces in unterschiedlichen Dateien unterstützt. Komfortabel ist auch die einfache Navigation zwischen Elementen und ihren Typen.

Quelle: <http://xsom.java.net/>

### iText

Für die Generierung der PDF-Dokumentation wird das iText-Framework in der Version 5.0.5 verwendet.

Quelle: <http://itextpdf.com/>

## Packages

### de.drv.dsrv.xtt.gui

Hauptelement in diesem Package ist die JFrame-Klasse *XsdCreator*, die alle relevanten Ein- und Ausgabe-Elemente besitzt (siehe Abbildung 1: Oberfläche XSD Profiling Tool). Daneben gibt es noch die beiden JDialog-Klassen *About* und *HelpDialog.*

Hauptelement zur Darstellung und Anpassung des XML-Schemas ist ein JTree-Objekt mit Checkboxen. Für die Darstellung und Eingabeverarbeitung sind die Klassen *CheckBoxNodeEditor* und *CheckBoxNodeRenderer* verantwortlich.

Für die Datenhaltung des JTrees ist ein eigenes TreeModel im folgenden Package implementiert.

### de.drv.dsrv.xtt.gui.model

Die Klasse *ProfilingTreeModel* implementiert ein TreeModel für die Repräsentation eines XML-Schemas im JTree. Dabei repräsentieren die Knoten jeweils ein Schemaelement. Alle für die Profilierung der Schemaelemente relevanten Informationen werden in den einzelnen Knoten gespeichert, wie die Kardinalitäten, Anmerkungen zum Element, ob das Element optional ist usw.

### de.drv.dsrv.xtt.tester

Die Klasse *Main* ist eine Konsolenanwendung zum automatisierten Testen der Profilierung.

### de.drv.dsrv.xtt.util

Dieses Package enthält die zentralen Logikklassen *ExtraTailoring* und *XsdCreatorCtrl* mit deren Implementierungen (*ExtraTailoringImpl*, *XsdCreatorCtrlImpl*) und Exception-Klassen (*ExtraTailoringException*, *XsdCreatorCtrlException, ValidationException*).

*ExtraTailoring* erzeugt aus einem vorgegebenen TreeModel eine Profilkonfiguration im XML-Format und generiert für eine gegebene Profilkonfiguration das entsprechende profilierte XML-Schema.

*XsdCreatorCtrl* ist eine Controller-Klasse mit der Steuerung der Ein- und Ausgaben der Benutzeroberfläche.

### de.drv.dsrv.xtt.util.pdf

*PdfCreator* bzw. die Implementierung *PdfCreatorImpl* ist zuständig für die Erzeugung einer PDF-Dokumentation für ein XML-Schema.

### de.drv.dsrv.xtt.util.schema

Dieses Package enthält alle Klassen zum Parsen und Erzeugen von XML-Schemadateien. Zentrale Klasse ist *MySchemaWriter* mit den Funktionen zum Serialisieren eines XML-Schemas.

### de.drv.dsrv.xtt.gui.util.tools

Die Klasse *Configurator* ist für den Zugriff auf die Einstellungen, die in den \*.properties-Dateien gespeichert sind, verantwortlich und enthält einige Package übergreifende Hilfsfunktionen.

Für das Lesen, Schreiben und Validieren von XML-Dateien enthält die Klasse *XsdXmlHelper* alle notwendigen Funktionen.

# Konfiguration und Anpassung

## Anwendungseinstellungen

Die Datei *xtt\_user.properties* enthält vom Benutzer anpassbare Einstellungen.

## Anmerkungen

In der Datei *xtt\_anmerkungen.properties* können Anmerkungsvorlagen zu den einzelnen Schemaelementen hinterlegt werden. Beim Erstellen einer neuen Profilkonfiguration werden diese Vorlagentexte übernommen und können in der Oberfläche ergänzt werden.