

# XSD Resolver

## *Handleiding*

**Voor intern gebruik**

Versie: 0.1

## Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>4</b>
<b>2</b>	<b>Techniek</b>	<b>4</b>
2.1	XSLT Stylesheets	4
2.2	Folder structuur	5
2.3	Configuratiebestand	6
2.3.1	application:processingParameters	6
2.3.2	application:standardApplicableMessages	6
2.3.3	application:namespacePrefixes	7
2.3.4	application: configurationSectorModel	7
<b>3</b>	<b>Processing</b>	<b>7</b>
3.1	Processor	7

## Versiebeheer XSD Resolver

**Auteur:** Robert Melskens - KING

Versie	Datum	Toelichting
0.1	11-12-2012	Initiele versie
0.2	12-04-2013	Uitleg over de structuur van het configuratiebestand opgenomen.

## 1 Inleiding

De XML-Schema's waaruit StUF berichtencatalogi en koppelvlakken bestaan zijn in de vorm waarin ze beheerd worden, tevens de genormaliseerde vorm, niet optimaal te gebruiken binnen de ontwikkelomgevingen van de diverse leveranciers. Zo bevat het constructies die het genereren van code bemoeilijken zo niet zelfs onmogelijk maken, met name het voorkomen van restrictions levert veel problemen op. Daarnaast bevatten de schema's componenten die niet gebruikt worden binnen de StUF berichtencatalogus of koppelvlak op basis waarvan dus bijv. onnodige code gegenereerd wordt. Ook het grote aantal bestanden binnen 1 namespace wil voor sommige software nog wel eens voor problemen zorgen. Met name import en include constructies leveren dan problemen op.

Om dit soort problemen op te lossen zonder het beheer van deze StUF berichtencatalogi en koppelvlakken te bemoeilijken is de XSD Resolver in het leven geroepen.

Het draagt zorg voor:

- Het plaatsen van alle componenten van 1 namespace binnen 1 XML-Schema;
- Het (desgewenst) omzetten van restriction constructies naar extension constructies;
- Het verwijderen van ongebruikte componenten uit de XML-Schema's.

In dit document wordt kort uitgelegd hoe de applicatie in elkaar zit (zie hoofdstuk 2) en hoe er mee omgegaan moet worden om tot een goed resultaat te komen (zie hoofdstuk 3).

## 2 Techniek

De applicatie draait in feite om een XSLT stylesheet waarmee de XML-Schema's worden geconverteerd. De folder structuur waarbinnen dit gebeurt en de batch bestanden die dit proces aansturen zijn echter onlosmakelijk met dit XSLT stylesheet (in feite een set van XSLT stylesheets) verbonden.

### 2.1 XSLT Stylesheets

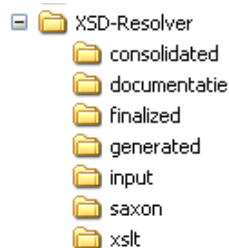
De relaties tussen de verschillende XSLT stylesheets, welk stylesheet wordt waar geïmporteerd, wordt hieronder geschetst:



Het stylesheet 'resolveXSDs.xslt' wordt bij de conversie aangeroepen en importeert via-via de andere stylesheets. De behoefte om de stylesheets op te splitsen in aparte stylesheets ontstond door het groeien van het hoofdstylsheet en het daardoor toenemen van de complexiteit. Opsplitsing gaf meer duidelijkheid m.b.t. de relatie tussen de verschillende templates en de verschillende stadia van het verwerkingsproces.

## 2.2 Folder structuur

De folderstructuur is onlosmakelijk verbonden met de XSLT stylesheets en ziet er als volgt uit:

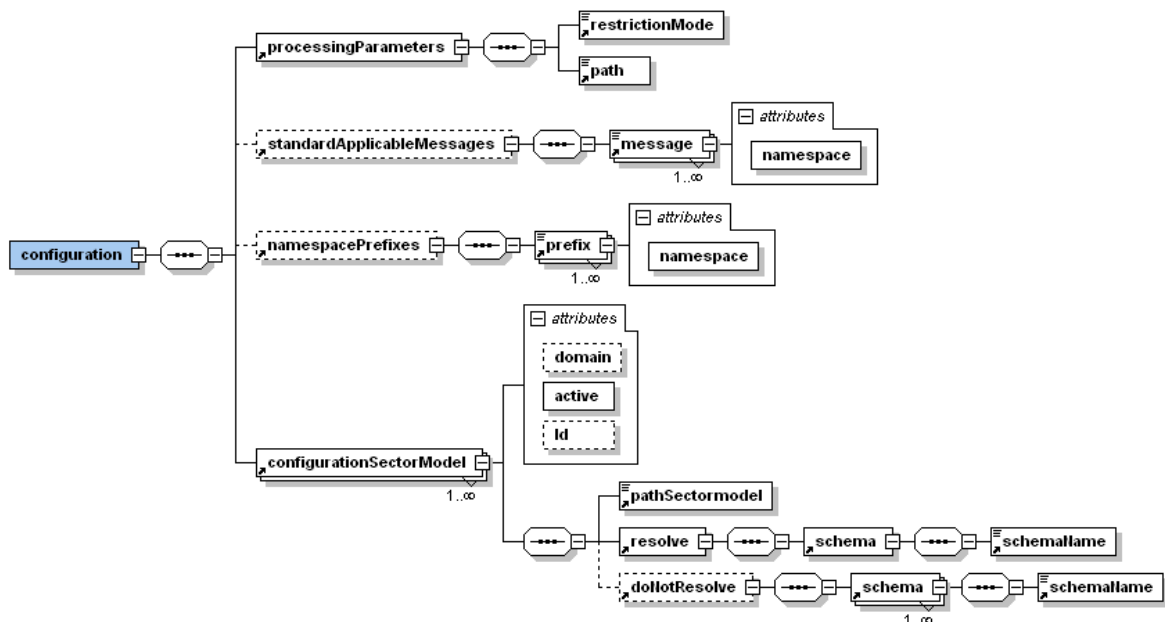


De folder 'XSD-Resolver' bevat naast de hier getoonde subfolders ook het batch bestand voor het opstarten van de applicatie. De subfolders hebben de volgende functie:

- de eerder genoemde XSLT stylesheet kunt u terugvinden in de folder 'xslt';
- het configuratiebestand waarmee het gehele proces wordt aangestuurd staat in de 'input' folder. In de volgende paragraaf **Fout! Verwijzingsbron niet gevonden.** gaan we even dieper in op het configuratiebestand;
- in de folder 'generated' vindt u de bestanden terug die gegenereerd worden bij het inventariseren van alle in de StUF berichtencatalogus of koppelvlak gebruikte XML-Schema's. Het bevat dus tussenresultaten;
- de folder 'consolidated' bevat alle geconsolideerde XML-Schema bestanden en bevat dus eveneens tussenresultaten;
- de folder 'finalized' bevat tenslotte de platgeslagen XML-Schema bestanden, het uiteindelijke doel van de XSD-Resolver applicatie;
- De onderliggende handleiding vindt u terug in de folder 'documentatie';
- In de folder 'saxon' vindt u de Saxon XSLT processor waarmee de XSLT stylesheets worden geprocessed. Deze folder of zelfs deze processor is in principe niet noodzakelijk op deze plaats. Het is nl. mogelijk om de processor op een geheel andere plaats in uw folderstructuur te plaatsen of zelfs om de processor geheel te vervangen door een andere XSLT processor. In beide gevallen zult u echter wel het batch bestand voor het opstarten van de applicatie moeten wijzigen.

## 2.3 Configuratiebestand

Het configuratiebestand is een XML bestand waarmee de XSLT stylesheets worden gevoed met procesinformatie. Zonder de informatie in dit bestand zijn de XSLT stylesheets van geen waarde. De structuur ziet er als volgt uit:



In de volgende paragrafen worden de elementen verder toegelicht.

### 2.3.1 application:processingParameters

Binnen 'application:processingParameters' element worden enkele processing parameters beschreven die noodzakelijk zijn voor de werking van de stylesheets.

- Ten eerste kan met het element 'application:restrictionMode' aangegeven worden of alle 'xs:restriction' elementen geresolved moeten worden of juist niet. Met resolvable bedoelen we dat daar waar een 'xs:restriction' constructie voorkomt binnen de context van een 'xs:complexType' dat element er samen met het omvattende 'xs:complexContent' element eruit gefilterd wordt. De content van het 'xs:restriction' element wordt dus in feite 2 niveaus hoger getild. In andere situaties kan een 'xs:restriction' constructie ook omgebouwd worden naar een 'xs:extension' constructie.
- Het 'application:path' element definieert de locatie van de XSD-Resolver applicatie zodat daarmee binnen de stylesheets de locaties van folders kan worden bepaald waarin de (tussen)resultaten kunnen worden geplaatst.

### 2.3.2 application:standardApplicableMessages

Standaard worden alleen de berichten meegenomen die gedefinieerd zijn in het mainschema van de te verwerken berichtencatalogus of koppelvlak. Omdat het daarnaast gewenst kan zijn ook berichten uit andere namespaces op te nemen in de te genereren schema's is dat met 'application:standardApplicableMessages' mogelijk gemaakt. Deze berichten worden dan in het element 'application:message' gedefinieerd tevens wordt de namespace uri gedefinieerd van de namespace waarin het bewuste bericht te vinden is. Voorbeelden van dit soort berichten zijn Fout- en Bevestigingsberichten.

### 2.3.3 application:namespacePrefixes

In een van de fases van het verwerkingsproces worden tijdens het processen van een berichtencatalogus of koppelvlak alle schema's binnen 1 geconsolideerd XML bestand geplaatst. Daarvoor worden alle unieke namespace declaraties gerepliceerd waarbij het voor kan komen dat de programmatuur 2 of meer default namespaces tracht aan te maken. Deze situatie leidt tot een conflict welke we oplossen door m.b.v. het element 'application:prefix'. In dit element wordt dan een prefix voor de betreffende namespaces gedefinieerd terwijl in het attribuut 'namespace' de betreffende namespace wordt gedefinieerd.

### 2.3.4 application: configurationSectorModel

Tenslotte moet voor een te verwerken sectormodel of koppelvlak een 'application: configurationSectorModel' element worden aangemaakt. Het element definieert de parameters voor het te verwerken sectormodel of koppelvlak.

Een sectormodel of koppelvlak zal wellicht vaker dan eens verwerkt moeten worden en het zou vervelend zijn steeds opnieuw dit element te moeten definiëren. Vandaar dat dit element meerdere keren kan voorkomen en m.b.v. het attribuut 'active' in of uitgeschakeld kan worden door het de waarde 'yes' of 'no' te geven. Alleen het eerste 'application:configurationSectorModel' element dat actief is wordt verwerkt. Het 'domain' en 'Id' attribuut hebben voorts nog geen functie anders dan documentatie.

Met het 'application:pathSectorModel' element kan aan de applicatie worden aangegeven in welke folder het te verwerken schema gevonden kan worden terwijl met het 'application:resolve/application: schema/application:schemaName' element de naam van het te verwerken schema opgegeven kan worden.

Tenslotte kan met het 'application:doNotResolve' element worden aangegeven indien een bepaalde schema's niet geresolved hoeft te worden.

Hiervoor kan het element 'application:schema' een of meerdere keren, al naar gelang het aantal schema's dat u niet wil resolvable, geplaatst worden.

Binnen het 'application:schema' element plaatst u vervolgens de elementen 'application:namespace' en 'application:schemaName'. In de eerste plaatst u de namespace van het schema dat u niet wil resolvable en in het tweede de naam van dat schema.

## 3 Processing

Het verwerken van het gewenste sectormodel gebeurt met behulp van het batch bestand 'XSD-Resolver.bat'. Voorafgaande daaraan zult u eerst het configuratiebestand moeten instellen, zie daarvoor het voorgaande hoofdstuk.

Het resultaat van het processen wordt in de folder 'finalized' geplaatst.

De gegenereerde schema bestanden bevatten voor de extensie altijd de toevoeging '\_resolved' zodat de originele schema bestanden van de gegenereerde onderscheiden kunnen worden.

Daarnaast kan in het batch bestand 'XSD-Resolver.bat' aan de parameter 'versie' een waarde worden toegekend. Deze waarde wordt dan aan de naam van de te genereren schema's tussen de extensie en '\_resolved' toegevoegd.

Het toekennen van een waarde gebeurt eenvoudig door achter 'versie=' de gewenste string in te geven, dus zonder quotes of doublequotes. De gewenste string mag geen spaties bevatten en natuurlijk alleen karakters die in een bestandsnaam zijn toegestaan.

### 3.1 Processor

De XSD-Resolver is standaard uitgerust met Saxon-HE 9.4.0.7J. Deze versie van Saxon vereist Java 5 (ook wel bekend als JDK 1.5) of later. Verwerking kan echter ook met andere processors gebeuren waaronder Saxon 9.4 voor .NET welke het .NET framework 2.0 of later vereist.

Zie voor meer informatie over Saxon '<http://saxon.sourceforge.net/>'

Indien besloten wordt een andere processor te gebruiken of de Saxon processor in een andere folder te plaatsen dan zal het batch bestand 'XSD-Resolver.bat' aangepast moeten worden of zal wellicht gekozen moeten worden voor een andere verwerkingsmethodiek.