

April 2024

Welkom bij deze eerste nieuwsbrief. Op deze plek gaan we in op opvallende veranderingen die in Invertor en de EA add-in zijn doorgevoerd, en hopen we Invertor gebruikers te helpen en te inspireren om nog meer te doen met Invertor.

[Alle nieuwsbrieven zijn hier terug te lezen.](#)

In deze nieuwsbrief:

Nieuwe release van Invertor OS.....	1
JSON schema's	1
SKOS volgens de SBB standaard.....	2
Vergelijken van versies van je model, en "release notes"	3
Waarschuwingen of fouten?	4
Plaatjes in de lopende tekst	4
Status van het "dashboard"	6
<i>Over deze nieuwsbrief</i>	6

Nieuwe release van Invertor OS

De ontwikkelversie van Invertor is sinds 4 april overgegaan in Invertor versie 4.0, dus een nieuwe *master* versie. Deze kun je nu kiezen binnen de Enterprise Architect add-in van Armatiek voor "ChainTranslateAndReport:4.0". Een overzicht van wat er is veranderd [vind je hier](#). Veel van de onderstaande informatie heeft betrekking op die nieuwe versie 4.0.

JSON schema's

Sinds enige tijd ondersteunt Invertor het genereren van JSON schema's (zie [GitHub](#)). Deze schema's beschrijven de mogelijke opbouw van JSON documenten. Zo'n schema kan worden ingeladen bij het

verwerken van JSON. Op die manier kan op tijd de verwerking tegengehouden worden, omdat het onjuiste informatie bevat (*validatie*).

De JSON schema's worden gegenereerd op basis van een [Best Practice van OGC](#), ontwikkeld met aansturing vanuit Geonovum. Deze standaard is nog in ontwikkeling, maar de bedoeling is dat deze breed zal worden ondersteund.

De implementatie betreft alleen MIM modellen. Wanneer je MIM volgt kun je een schema laten samenstellen met de volgende *properties*:

```
# Maak MIM serialisatie
createmimformat = yes
mimformattype = xml
mimformatversion = v2
# Maak een Json schema
createjsonschema = yes
jsonschemasource = MIM
createjsonschemavariant = plainjson
```

De eerste drie properties zorgen ervoor dat een MIM serialisatie wordt gemaakt. Deze vormt de basis van de JSON schema serialisatie.

In de laatste drie properties wordt aangegeven welk type JSON schema je wilt maken. De laatste property kan drie waarden hebben, voor drie schema conventies: `plainjson` (Plain JSON), `geojson` (GeoJSON) of `jsonfg` (JSON -FG).

De standaard heeft meerdere opties om tot een gewenste vorm van de schema's te komen (zgn. *requirement classes*). Een organisatie zal één of enkele van zulke combinaties van opties ondersteunen. In de GitHub wordt voor iedere organisatie bijgehouden welke combinaties gewenst zijn. Neem contact op met Armatiek als je deze nieuwe functie wil gaan gebruiken.

Beschikbaar in Invertor 4.0 en de Nightly-build.

SKOS volgens de SBB standaard

De [standaard voor het beschrijven van begrippen](#) (SBB) geeft aan hoe begrippen in een begrippenlijst, taxonomie of thesaurus eenduidig worden beschreven. Deze afspraken kunnen worden opgenomen in een SKOS (*linked data*) specificatie.

In informatiemodellen komen nogal wat “begrippen” voor, en deze kun je aan het model onttrekken in de vorm van een SKOS document. Dit kan een goede start vormen voor de opbouw van een begrippencatalogus.

In Invetor kun je deze SKOS genereren met de property:

```
createskos = yes
```

Het resultaat is een Turtle (TTL) file met de SKOS info. Per organisatie kan worden afgesproken welke info daarin terechtkomt. Een basis is reeds gelegd door SBB.

Beschikbaar in Invetor 4.0 en de Nightly-build.

Vergelijken van versies van je model, en “release notes”

Wanneer je een model verwerkt met Invetor is het soms aantrekkelijk de verschillen te zien met een vorige versie van het model, of met het model waarop jouw model is gebaseerd (de zgn. *supplier*).

De release van een informatiemodel geef je op als tagged value “Release”. Het heeft doorgaans de vorm van een jaar, maand en dag: YYYYMMDD. Als je jouw model wilt vergelijken met een vorige release, bijvoorbeeld release 20240301, dan geef je op:

```
# Vergelijk met een eerdere release
compare = release
comparewith = 20240301
```

Dit resulteert in een overzicht van alle verschillen, als onderdeel van het rapport. Bijvoorbeeld:

Comparison report

This report shows 10 differences found between models `SIM/GitHub-issues-456-compare/20240301` and `SIM/GitHub-issues-456-compare/20240302`.

#	Diff	Domain	Class	Att/Assoc	Property	Old	New
00001	Changed				Release (tagged value)	20240301	20240302
00002	Added				Supplier-name (tagged value)		GitHub-issues-456-compare
00003	Added				Supplier-project (tagged value)		SIM
00004	Added				Supplier-release (tagged value)		20240301
00005	Changed	Model (domain)			derived	false	true
00006	Changed	Model (domain)			Release (tagged value)	20240301	20240302
00007	Added	Model (domain)	Haarkleur (enumeratie)	Donker (enumeratiewaarde)			
00008	Removed	Model (domain)	Haarkleur (enumeratie)	Zwart (enumeratiewaarde)			
00009	Added	Model (domain)	School (objecttype)	adres (attribuutsoort)			
00010	Changed	Model (domain)	School (objecttype)	heeft leerling (relatiesoort)	min-occurs	1	3

Als je de `comparewith` parameter weglaat, wordt het model vergeleken met de meest recente versie van hetzelfde model.

Er zijn plannen om ook de *release notes* van een model te genereren op basis van deze vergelijking. Dat kan natuurlijk nooit complete release documentatie opleveren, maar een goede start is wel te maken. VNG heeft hiervoor [al eerder een voorstel gedaan](#).

Zie ook [verdere documentatie](#).

Beschikbaar in Invertor 4.0 en de Nightly-build.

Waarschuwingen of fouten?

Bij het valideren van een model wordt gekeken of verplichte *tagged values* een waarde hebben gekregen. In veel situaties zijn die waarden niet opgegeven: met name als een model in ontwikkeling is, is dat ook begrijpelijk. De bron van de definitie van een objecttype, of zelfs de definitie zelf, kan pas later in het proces worden ingevoegd. Het is niet wenselijk er steeds op gewezen te worden dat deze informatie mist.

Je merkt het bij het runnen van een concept model: de waarschuwingen vliegen je om de oren. Ze zijn terecht maar komen gewoonweg te vroeg in het proces. Als je ervan af wilt, is daarvoor een property beschikbaar:

```
validatederivation = no
```

In feite is het testen of *tagged values* wel correct zijn opgegeven, onderdeel van de “afleiding” van modellen. Pas als alle informatie die voor een model relevant is bekend is, kan worden vastgesteld of ook alle *tagged values* goed zijn ingevoerd. Want deze waarden *kunnen* van andere modellen worden overgenomen. Invertor neemt deze test dus op na het verzamelen van alle info (afleiding) omtrent het model. Vandaar deze op het eerste gezicht wat vreemde property naam.

Plaatjes in de lopende tekst

Het stond al enige tijd op de rol, maar het is nu ook beschikbaar: het opnemen van plaatjes (foto’s, schema’s e.d.) in de lopende tekst van het Respec document. Enterprise Architect ondersteunt het weergeven van plaatjes in het notatieveld nog niet. Maar het biedt wel de mogelijkheid een referentie naar een plaatje op te nemen. Dat gaat via de “Image manager”. In een notatieveld voeg je

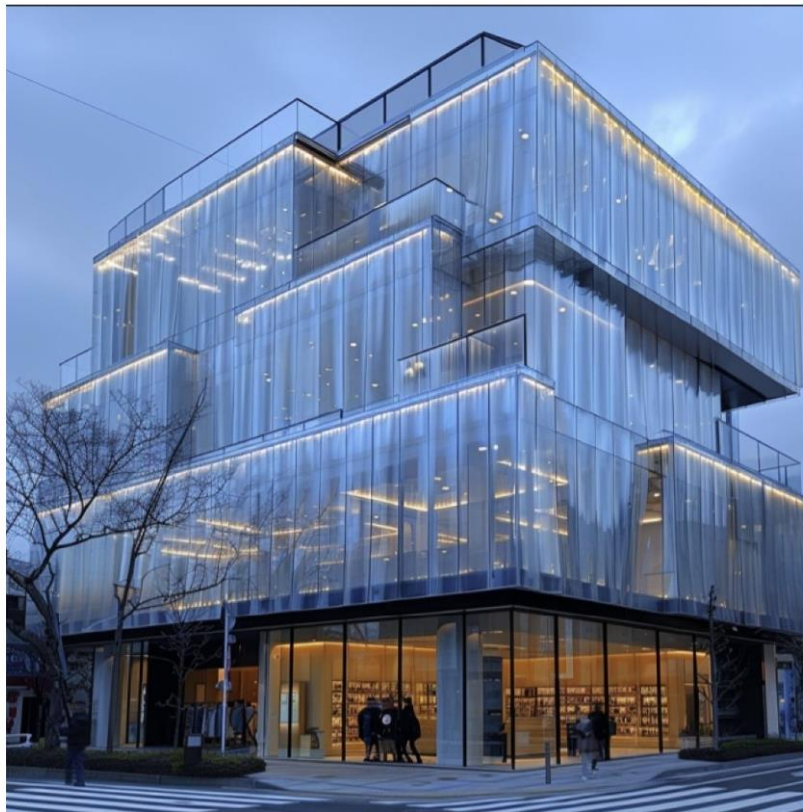
waar je een plaatje wilt opnemen een link in, waarna je kiest voor de “Image Manager” (in *hyperlink details*). Het plaatje wordt getoond als een link, met de naam van het bestand.

Bij het samenstellen van de Respec catalogus wordt het plaatje ingevoegd op de aangegeven plek. Dit is dus alleen beschikbaar in notatievelden die worden geïnterpreteerd als definitie, toelichting of andere geconfigureerde, “vrije tekst”-delen van het model.

Een voorbeeld is hieronder opgenomen.

§ 2.1.1 Glasgebouw

Naam	Glasgebouw
Definitie	Een glasgebouw is een gebouw waarvan de buitenkant voornamelijk bestaat uit glas.
Toelichting	Een voorbeeld van een glasgebouw is:



Het gebouw is maximaal transparant.

Voorwaarde voor het werken met deze plaatjes is dat de extensie van het bestand in de naam van het bestand blijft gehandhaafd. Dat ziet er in EA als volgt uit:

The screenshot displays the Invertor software interface. On the left, a class definition for «Objecttype» Glasgebouw is shown, including an attribute «Attribuutsoort» + adres: CharacterString. The main window shows the class name 'Glasgebouw' and a list of properties: General, Responsibilities, Requirements, Constraints, Scenarios, Files, and Related Links. The text content is displayed in a rich text editor with a toolbar. The text includes: 'Een glasgebouw is een gebouw waarvan de buitenkant voornamelijk bestaat uit glas.', a comment '-- Toelichting --', 'Een voorbeeld van een glasgebouw is:', a link 'test4.png', and 'Het gebouw is maximaal transparant.'

Deze functie is ook beschikbaar in de MsWord HTML export (ten behoeve van een PDF export). Hierbij moeten de plaatjes wel handmatig geschaald worden.

Beschikbaar in Invertor 4.0 en de Nightly-build.

Status van het “dashboard”

De SaaS biedt naast de Enterprise Architect *add-in* een dashboard dat onder andere toegang geeft tot jobs en modellen die met Invertor OS zijn gemaakt. Dit dashboard wordt relatief weinig gebruikt en vraagt relatief veel onderhoud. Armatiek Solutions heeft besloten het dashboard zoals we dat nu kennen in de toekomst niet te continueren. We zijn in overleg met de gebruikers om vast te stellen hoe we de belangrijkste functionaliteiten kunnen handhaven. Doelstelling is de komende jaren een complete en robuuste dienst te kunnen blijven aanbieden.

Over deze nieuwsbrief

De Nieuwsbrieven Invertor worden verspreid onder alle geregistreerde gebruikers van Invertor SaaS van [Armatiek Solutions BV](#). Vanaf april 2024 komen minimaal 4 nieuwsbrieven per jaar uit. [Alle zijn hier terug te lezen](#). Verbeteringen, correcties, suggesties, afmelden? Mail naar info@armatiek-solutions.nl.