

Leeswijzer Assertions

Met ingang van de nieuwe versie van de FTK en PPI taxonomie is de indeling van het assertions tabblad gewijzigd. Dit document geeft een korte toelichting op de syntax van de assertions.

Op het assertions tabblad worden nu de volgende gegevens getoond:

1. **Module(s)** De module(s) waarin de validatieregel wordt gebruikt
2. **ID** Het ID van de validatieregel
3. **Changed per** Periode waarin de validatieregel is gewijzigd
4. **Last change** Versie van de taxonomie van de laatste wijziging
5. **Threshold** Toegestane foutmarge
6. **Severity** Het wel of niet blokkerend zijn van een validatieregel
7. **Tables** De tabel(len) die in de validatieregel worden gebruikt
8. **Formula** De uit te voeren validatie
9. **Label (nl)** Bericht bij het niet voldoen aan de validatieregel

De kolommen waarin in het verleden rijen, kolommen en/of sheets werden vermeld waarop de validatieregel als "pattern" worden toegepast zijn verplaatst naar de kolom "Formula".

De kolom 'expression' is komen te vervallen, dit was een technische weergave van de formula, deze expression wordt nog wel getoond in het validatierapport dat na uitvoeren van de validatieregels wordt opgeleverd.

De specificaties in de "Formula" kolom zijn in lijn gebracht met het tool waarmee de validatieregels zijn geïmplementeerd. De syntax wijkt iets af van de eerdere validatieregels, constructies als r[010-020] zijn vervangen door r010:r020.

Syntax van formulas:

Er zijn twee soorten formula's:

- Enkelvoudige formulas die op vaste rijen, kolommen en/of sheets worden uitgevoerd.
- "For each" formulas die meerdere keren op rijen, kolommen en/of sheets worden uitgevoerd (deze vervangen voor een deel de patterns uit eerdere specificaties, andere patterns zijn gesplitst in afzonderlijke validaties te herkennen aan een toevoeging a, b, c etc. aan het ID).

De enkelvoudige formulas hebben een syntax in de vorm van:

a=b, a=sum(b), if (a=b) then (c>d) else true() etc.

Bij "For each" formulas is de structuur:

[tabellem,rijen,kolommen en/of sheets] -> [formula]

Hierbij wordt de formula uitgevoerd voor de tabellen, rijen, kolommen en/of sheets die tussen de eerste [] zijn gespecificeerd. In de formula worden ontbrekende tabellen, rijen, kolommen en/of sheets in de formula ingevuld. Bijvoorbeeld:
[t01.01, r010,r020,r030] -> [{c030} >= {c040}]

De formula: {c030}>={c040} wordt voor tabel t01.01 voor de rijen r010, r020 en r030 uitgevoerd.

De "for each" constructie is soms ook gebruikt om de formula compacter te kunnen schrijven als er veel velden met klaar worden vergeleken, bijvoorbeeld:

[t01.01] -> [({c010, r020} = {c080, r010}) and ({c010, r030} = {c080, r020}) and ({c010, r040} = {c080, r030})]

In deze formula wordt het tabelnummer aan alle gespecificeerde cellen toegevoegd.

Rijen en kolommen kunnen als enkele rijen/kolommen of ranges van rijen/kolommen worden gespecificeerd:

- Enkele rijen/kolommen worden gescheiden door een komma, bijvoorbeeld:
r010,r020,r030
- Ranges worden gescheiden door een dubbele punt bijvoorbeeld:
r010:r090
- Combinaties van enkele rijen/kolommen en ranges zijn ook toegestaan, bijvoorbeeld:
r010,r030,r050:r080,r100

Wanneer een enkele cel wordt gespecificeerd staat deze tussen enkele accolades { }, wanneer een range van cellen moet worden gebruikt, wordt deze tussen dubbele accolades gespecificeerd {{ }}.

Bij het controleren of een enkele cel leeg is wordt dit gespecificeerd als:
empty({t01.01,r020,c030}), moet er worden gecontroleerd of één van de cellen in een range leeg is dan wordt het als volgt gespecificeerd:
empty({{t01.01,r010:r030,c030}})

Bij functies als sum() en count() moet binnen de haakjes een range worden gedefinieerd, als bijvoorbeeld cellen uit verschillende tabellen moeten worden opgeteld, dan zijn extra () gebruikt: bijvoorbeeld
sum({t01.01,r010,c010},{t01.02,r030,c010}).

Wanneer een formula op alle rijen, kolommen of sheets moet worden uitgevoerd wordt dit gespecificeerd met rNNN, cNNN of sNNN. Bij open-as tabellen wordt de formula automatisch op alle sheets, rijen of kolommen uitgevoerd.

Wanneer een veld aan een waarde uit een lijst van toegestane waarden moet voldoen is dit gedefinieerd als {t01.01,r010,c010}=('waarde1', 'waarde2', 'waarde3')

Vergelijkingstekens als >, >=, <= kunnen ook worden gespecificeerd als gt, ge, le niet gelijk aan wordt geschreven als != of ne.

Bij een if ... then .., else validatie wordt in de else vaak true() gedefinieerd; dit zorgt ervoor dat de formula niet af gaat als de conditie waarmee wordt getest niet van toepassing is.

In de formula zijn waarden uit keuzelijstjes als leesbare tekst gepresenteerd
bijvoorbeeld:

```
if ({{t01.01,r110,c010}}='Anders') then (not(empty({t01.01,r070,c010}))) else true()
```

In de expressie die in een validatierapport wordt getoond is de tekst 'Anders' vervangen door de code van het domein en de member van het keuzelijstje dat aan de cel is gekoppeld, dit wordt dan bijvoorbeeld:

```
if ($a=xs:QName('dnb_ftk_dom006:mem012')) then (exists($b)) else true()
```

Threshold

De threshold is af te leiden uit de gebruikte functies, wanneer een custom margin is toegestaan zijn in de formula functies gebruikt die beginnen met cmf:

Bijvoorbeeld: cmf:numeric-equal({r190}, sum({{r140:r180}}), 1000)

De eerste twee parameters van de functie zijn de waarden die met elkaar worden vergeleken, de laatste parameter is de toegestane marge, in het voorbeeld 1000.

Wanneer interval-arithmetic wordt gebruikt om de threshold te bepalen dan beginnen de functies met iaf:, bijvoorbeeld iaf:sum({{r010:r100}})

Bijzonderheden

Een aantal validaties wordt uitgevoerd op rijen en/of kolommen die zowel bedragen als andere datatypes bevatten (percentages, tekst e.d). De software die wordt gebruikt om de validaties aan te maken geeft waarschuwingen als dit soort datatypes door elkaar worden gebruikt. Om deze reden zijn sommige validaties gesplitst in een validatie voor bedragen en een validatie voor overige velden.