

Handleiding SLD's in Geoserver

Bijgewerkt: 17-8-2017

Inleiding

Geoserver is een open source applicatie voor het maken van WMS'en en WFS'en, welke wordt onderhouden door een grote online community en geleverd wordt door B3Partners. Online is veel Engelse documentatie te vinden over de werking van Geoserver en de verschillende functies. WMS'en welke gemaakt worden met Geoserver maken voor de verbeelding van de data in een kaartlaag gebruik van een Styled Layer Descriptor, een SLD.

Een SLD is een bestandje met daarin een code die de eigenschappen van een kaartlaag bevat. Voor elke kaartlaag wordt dus één SLD gebruikt. Online is veel te vinden over de werking van SLD's en hoe deze ingericht moeten worden in het zogenaamde 'SLD Cookbook' (<http://docs.geoserver.org/stable/en/user/styling/sld-cookbook/>).

Het SLD Cookbook is alleen in het Engels opgesteld en ook dusdanig uitgebreid dat het af en toe lastig is om hetgeen te vinden dat de beheerder nodig heeft. Daarom worden in deze handleiding enkele veel voorkomende wensen met betrekking tot de verbeelding van data via SLD's besproken en toegelicht.

Aangezien meestal van een Engelse óf Nederlandse installatie van Geoserver sprake is en de taal ook browserafhankelijk kan zijn, zullen sommige termen dubbel gebruikt worden. Bijvoorbeeld: 'Workspace/omgeving', hier wordt in twee talen dezelfde functie bedoeld.

SLD's

Voordat gewerkt wordt met SLD's is het handig om enig inzicht te hebben in de manier waarop een SLD is opgebouwd. De structuur van een SLD bestaat uit verschillende elementen, beginnend met de StyledLayerDescriptor (<StyledLayerDescriptor>). Binnen StyledLayerDescriptor vallen de elementen <Layer>, waarin de laag wordt gedefinieerd en <Style>, waarin de stijl daadwerkelijk wordt gedefinieerd.

Binnen styles vallen rules (< Rule>). Hierin wordt de daadwerkelijke verbeelding van de attributen gedefinieerd. Eerst wordt er aangegeven wat de naam van de rule is, vervolgens op welke attributen de regel slaat (Te herkennen aan de term 'PropertyIs [...]') en wat voor verbeelding er gebruikt moet gaan worden met een aantal specificaties voor die verbeelding (kleur, grootte, doorzichtigheid, labels, etc.). Er kunnen meerdere rules gedefinieerd worden.

Alle elementen moeten worden afgesloten met een '/' ->

<elementnaam>waarde</elementnaam>

Een standaard SLD voor een rood, vierkant punt ziet er zo uit:

```

1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <StyledLayerDescriptor version="1.0.0"
3      xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
4      xmlns="http://www.opengis.net/sld"
5      xmlns:ogc="http://www.opengis.net/ogc"
6      xmlns:xlink="http://www.w3.org/1999/xlink"
7      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8      <NamedLayer>
9          <Name>Default Point</Name>
10         <UserStyle>
11             <Title>A boring default style</Title>
12             <Abstract>A sample style that just prints out a red square</Abstract>
13             <FeatureTypeStyle>
14                 <Rule>
15                     <Name>Rule 1</Name>
16                     <Title>RedSquare</Title>
17                     <Abstract>A red fill with an 11 pixel size</Abstract>
18                     <PointSymbolizer>
19                         <Graphic>
20                             <Mark>
21                                 <WellKnownName>square</WellKnownName>
22                                 <Fill>
23                                     <CssParameter name="fill">#FF0000</CssParameter>
24                                 </Fill>
25                             </Mark>
26                             <Size>6</Size>
27                         </Graphic>
28                     </PointSymbolizer>
29                 </Rule>
30             </FeatureTypeStyle>
31         </UserStyle>
32     </NamedLayer>
33 </StyledLayerDescriptor>

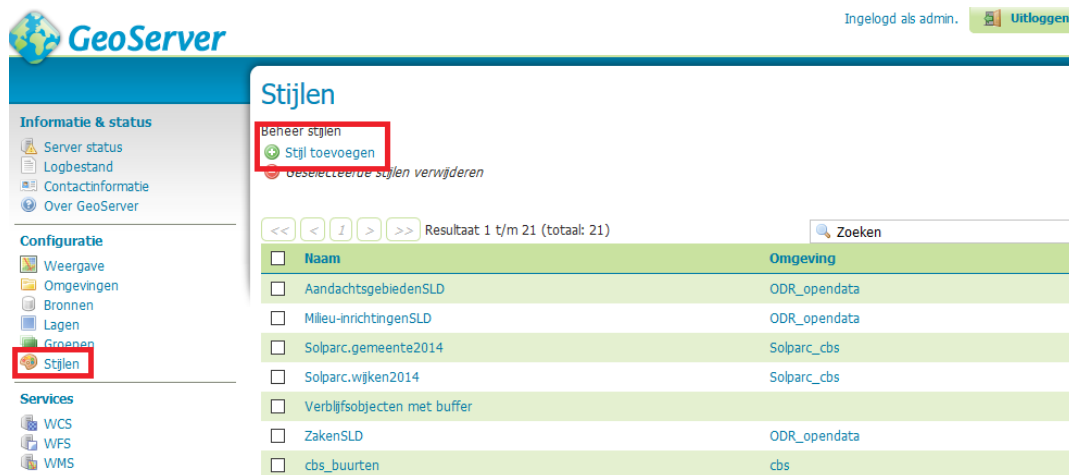
```

SLD's toevoegen

Voor het maken van een SLD kunnen de volgende stappen gezet worden.

Stap 1

Log in in Geoserver en klik links op 'Stijlen/Styles' onder 'Configuration/Configuratie'. Klik vervolgens op 'Stijl toevoegen'.



Stap 2

Vul de rood omlijnde stukken onder in de afbeelding aan het einde van deze stap in. Er zijn hier nog een aantal andere mogelijkheden. Met de dropdown onder 'omgeving' kan de SLD aan een omgeving gekoppeld worden.

Met 'Generate a default style' (versie 2.8.x) kan een simpele standaard SLD gegenereert worden (een gekleurde point, line of polygon, een raster SLD of een algemene SLD, welke van toepassing kan zijn op alle soorten lagen). Dit is vooral bruikbaar bij tests of als er om een andere reden snel een tijdelijke SLD nodig is.

Met 'kopiëer een bestaande stijl' kan een bestaande stijl overgenomen worden. Dit is handig als de beheerder een stijl heeft welke in meerdere situaties bruikbaar is, maar bijvoorbeeld een andere kleur gebruikt moet worden. Er hoeft dan alleen maar een kleine aanpassing gedaan te worden in het duplicaat.

Met de knop 'Bladeren' kan de beheerder een bestaande SLD vanaf de eigen computer uploaden. De knop 'Valideren' wordt gebruikt om te kijken of de SLD geen cruciale stukken mist. Met 'Toepassen' wordt de SLD

opgeslagen. 'Legenda bekijken' toont de legenda afbeelding ter controle.

In het tekstblok wordt het SLD gemaakt. Dit kan op verschillende manieren. De beheerder kan de SLD zelf typen, met bijvoorbeeld een bestaande SLD als een begin om deze zelf makkelijk te bewerken. Kleuren worden in SLD's aangegeven in Hexadecimale kleuren, bijvoorbeeld #90C3D4, of #FFFFFF.

Aan het einde van dit document vindt u twee pagina's met in SLD's gebruikte elementen.

Een andere manier is door in het (aparte) programma Qgis een laag te maken en te verbeelden, om vervolgens dit op te slaan als .sld (te vinden onder 'Opslaan als'). Deze SLD kan dan de beheerder dan uploaden. Qgis gebruikt in de SLD alleen met regelmaat `<se:[operatornaam]>` en `</se:[operatornaam]>`. Het stukje 'se:' moet dan veranderd worden naar 'sld:'.

Nieuwe stijl

Voer een nieuwe SLD-definitie in, gebruik een bestaande als een sjabloon, of upload een voorbereerde stijl vanaf je bestandssysteem. Je editor verzorgt syntax highlighting en kan schermvullend gebruikt worden. Klik op de knop "valideren" om te verifiëren dat je stijl een geldig SLD document is.

Naam

Omgeving

Formaat

Generate a default style

Kopieer van bestaande stijl

bestand met stijlen

Bladeren... Geen bestand geselecteerd. Uploaden ...

Valideren Legenda bekijken Toepassen Annuleren

Foutmeldingen wanneer een SLD wordt 'toegepast' worden vaak voorzien van een regelnummer. Dit geeft aan waar het probleem ongeveer ligt.

SLD's voor punten

Een simpele punt SLD (zonder begin en eind van het volledige SLD) ziet er als onderstaande afbeelding uit.

```
1  [ ] <FeatureTypeStyle>
2  [ ]   <Rule>
3  [ ]     <PointSymbolizer>
4  [ ]       <Graphic>
5  [ ]         <Mark>
6  [ ]           <WellKnownName>circle</WellKnownName>
7  [ ]           <Fill>
8  [ ]             <CssParameter name="fill">#FF0000</CssParameter>
9  [ ]           </Fill>
10 [ ]        </Mark>
11 [ ]       </Graphic>
12 [ ]     </PointSymbolizer>
13 [ ]   </Rule>
14 [ ] </FeatureTypeStyle>
```

Er zijn een aantal standaard symbolen (hier: <WellKnownName>) in geoserver welke gebruikt kunnen worden om een punt weer te geven:

- square
- circle
- triangle
- star
- cross
- x

Daarnaast zijn er nog een aantal andere symbolen die gebruikt kunnen worden:

- shape://vertline
- shape://horline
- shape://slash
- shape://backslash
- shape://dot
- shape://plus
- shape://time
- shape://oarrow
- shape://carrow

<http://docs.geoserver.org/stable/en/user/styling/sld-extensions/pointsymbols.html> (scroll naar 'Shape Symbols' voor uitleg)

Punten transparant maken

Transparantie wordt in een SLD gerealiseerd met de term 'opacity'. Vul in het stuk tussen <Fill> en </Fill>, onder de kleurcode het volgende in:

```
<CssParameter name="fill-opacity">0 tot 1</CssParameter>
```

Met daarbij 0 als volledig transparant en 1 helemaal niet transparant.

Punten met rand

Voeg aan de rule het volgende toe, na `</Fill>` en voor `</Mark>`:

```
<Stroke>
  <CssParameter name="stroke">Kleurcode</CssParameter>
  <CssParameter name="stroke-width">pixels</CssParameter>
</Stroke>Een transparante rand wordt gerealiseerd met:
<CssParameter name="stroke-opacity">0 tot 1</CssParameter>
```

Met daarbij 0 als volledig transparant en 1 helemaal niet transparant.

Punten draaien

Voeg aan de rule het volgende toe, direct voor `</Graphic>`:

```
<Rotation>Aantal_graden</Rotation>
```

Punten met label

Voeg aan de rule het volgende toe, na `</PointSymbolizer>` en voor `</Rule>`:

```
<TextSymbolizer>
  <Label>
    <ogc:PropertyName>attribuutnaam</ogc:PropertyName>
  </Label>
  <Font>
    <CssParameter name="font-family">Font</CssParameter>
    <CssParameter name="font-size">grootte</CssParameter>
    <CssParameter name="font-style">stijl</CssParameter>
    <CssParameter name="font-weight">Dikte</CssParameter>
  </Font>
  <LabelPlacement>
    <PointPlacement>
      <AnchorPoint>
        <AnchorPointX>0 tot 1</AnchorPointX>
        <AnchorPointY>0 tot 1</AnchorPointY>
      </AnchorPoint>
      <Displacement>
        <DisplacementX>pixels</DisplacementX>
        <DisplacementY>pixels</DisplacementY>
      </Displacement>
      <Rotation>-45</Rotation>
    </PointPlacement>
  </LabelPlacement>
  <Fill>
    <CssParameter name="fill">kleur</CssParameter>
  </Fill>
</TextSymbolizer>
```

In bovenstaande stuk zijn een aantal zaken van belang:

- Achter `<ogc:PropertyName>` volgt de exacte naam van het attribuut wat moet dienen als label
- Met `` wordt de stijl van de letters bepaald.
- Font-family = de naam van het font wat beschikbaar is voor Geoserver, bijvoorbeeld Arial
- Font-size = de grootte van de letters
- Font-style = 'normal', 'italic' of 'oblique'
- Font-weight = 'normal' of 'bold'
- `<Anchorpoint>` is dat punt ten opzichte waarvan het label verplaatst kan worden, het gaat hier om een intersect tussen het punt enerzijds en het label anderzijds. 0,5 betekent hier dat het centrum van het label horizontaal, danwel verticaal intersect met het punt.
- `<displacement>` werkt in pixels en geeft aan hoeveel pixels het label moet verplaatsen ten opzichte van het ankerpunt.
- `<Rotation>` geeft aan of het label gedraaid moet worden en zo ja, in hoeveel graden (niet verplicht).
- `<Fill>` bepaald hier de kleur van het Label.

Een transparant label wordt gerealiseerd met:

```
<CssParameter name="font-opacity">0 tot 1</CssParameter>
```

Met daarbij 0 als volledig transparant en 1 helemaal niet transparant.

Wanneer het de bedoeling is om ook een randje rond het label te projecteren moet gebruik gemaakt worden van `<Halo>`:

```
<Halo>
  <Radius>Pixels</Radius>
  <Fill>
    <CssParameter name="fill">Kleur</CssParameter>
  </Fill>
</Halo>
```

Zet dit voor `</TextSymbolizer>`.

Het is ook mogelijk om prioriteiten te geven op basis van attributen aan labels.

```
<Priority>
  <PropertyName>attribuut</PropertyName>
</Priority>
```

In bovenstaande voorbeeld wordt er door Geoserver een prioriteit aan labels gegeven van objecten met een grotere waarde in kolom `attribuut`

Zet dit voor `</TextSymbolizer>`.

Punten verbeelding op basis van attributen

Soms is het gewenst om verschillende waarden van een attribuut op verschillende manieren te tonen. Bijvoorbeeld grotere punten of punten met andere kleuren voor grotere waarden.

Wat in dit geval dan van belang is, is dat er voor elke waarde welke voortkomt uit het onderscheid wat gemaakt gaat worden een nieuwe <rule> in het SLD gemaakt wordt. In elke regel wordt er dan gefilterd op een andere waarde en wordt er een verbeelding meegegeven.

In het onderstaande voorbeeld wordt er een onderscheid gemaakt tussen grootte van bevolkingen. De verschillende grootten krijgen verschillende formaten mee. Na de <Rule> volgt een <ogc:PropertyFilter>, waarna de <PointSymbolizer> volgt.

```
<Rule>
  <Name>SmallPop</Name>
  <Title>Title</Title>
  <ogc:Filter>
    <ogc:PropertyIsLessThan>
      <ogc:PropertyName>Name</ogc:PropertyName>
      <ogc:Literal>Waarde</ogc:Literal>
    </ogc:PropertyIsLessThan>
  </ogc:Filter>
  <PointSymbolizer>
    <Graphic>
      <Mark>
        <WellKnownName>circle</WellKnownName>
        <Fill>
          <CssParameter name="fill">Kleurcode</CssParameter>
        </Fill>
      </Mark>
      <Size>Pixels</Size>
    </Graphic>
  </PointSymbolizer>
</Rule>
```

```
<Rule>
  <Name>MediumPop</Name>
  <Title>Title</Title>
  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsGreaterThanOrEqualTo>
        <ogc:PropertyName>Name</ogc:PropertyName>
        <ogc:Literal>Waarde</ogc:Literal>
      </ogc:PropertyIsGreaterThanOrEqualTo>
      <ogc:PropertyIsLessThan>
        <ogc:PropertyName>Name</ogc:PropertyName>
        <ogc:Literal>Waarde</ogc:Literal>
      </ogc:PropertyIsLessThan>
    </ogc:And>
  </ogc:Filter>
  <PointSymbolizer>
```

```
<Graphic>
  <Mark>
    <WellKnownName>circle</WellKnownName>
    <Fill>
      <CssParameter name="fill">Kleurcode</CssParameter>
    </Fill>
  </Mark>
  <Size>Pixels</Size>
</Graphic>
</PointSymbolizer>
</Rule>
```

Wat bij bovenstaande stuk van belang is, is dat er in het tweede stuk `<ogc:Filter>` gebruik gemaakt wordt van het element `<ogc:And>`. Dit geeft aan dat er gebruik gemaakt moet worden van alle filters in het stuk tussen `<ogc:And>` en `</ogc:And>`.

Er kan van de volgende 'binaire' filteroperators gebruik gemaakt worden:

- `<PropertyIsEqualTo>`
- `<PropertyIsNotEqualTo>`
- `<PropertyIsLessThan>`
- `<PropertyIsLessThanOrEqualTo>`
- `<PropertyIsGreaterThan>`
- `<PropertyIsGreaterThanOrEqualTo>`

Er zijn ook drie 'waarde' filteroperators te gebruiken:

- `<PropertyIsLike>*`
- `<PropertyIsNull>`
- `<PropertyIsBetween>`

Deze operators worden allen, wanneer gefilterd moet worden op de waarde van een attribuut, gevolgd door:

```
<PropertyName>attribuutnaam</PropertyName>
<Literal>attribuutwaarde</Literal>
```

* Het filter `<PropertyIsLike>` moet voorzien worden van:

wildCard="*" *singleChar*="#" *escapeChar*="!". Hierin is *wildCard* het teken wat staat voor 0 of meer karakters. *singleChar* is het teken wat staat voor één enkel willekeurig karakter. *escapeChar* staat voor het teken wat ervoor zorgt dat (in bovenstaand geval) gezocht kan worden op * en # als teken.

- **Verdieping*** → geeft alle waardes die beginnen met 'Verdieping'
- **Verdieping#** → geeft alle waardes die voldoen aan term 'Verdieping' en die gevolgd worden door 1 karakter.
- **Verdieping!#** → Geeft alle waardes die voldoen aan term 'Verdieping#' terug.

Punten schaalafhankelijk maken

Het is mogelijk om punten schaalafhankelijk te maken, dit is vooral van belang wanneer er veel punten getoond moeten worden. Het laden van veel punten in een kaart kan ervoor zorgen dat de kaart langzaam wordt. Daarom kan het van belang zijn ervoor te zorgen dat niet alle punten direct worden geladen, maar pas vanaf een bepaald niveau.

Hiervoor worden de elementen `<MinScaleDenominator>` en `<MaxScaleDenominator>` gebruikt (Min=Minimal, Max=Maximal). Dit wordt vóór de `<PointSymbolizer>` als volgt aangegeven:

```
<MaxScaleDenominator>getal</MaxScaleDenominator>
```

of

```
<MinScaleDenominator>getal</MinScaleDenominator>  
<MaxScaleDenominator>getal</MaxScaleDenominator>
```

of

```
<MinScaleDenominator>getal</MinScaleDenominator>
```

Deze scales gaan over de schaal van de kaarten. Wanneer er bijvoorbeeld zou staan:

```
<MaxScaleDenominator>50000</MaxScaleDenominator>
```

Betekent dit dat de regel alleen werkt wanneer de kaart een schaal van 1:50000 of minder heeft.

```
<MinScaleDenominator>50000</MinScaleDenominator>  
<MaxScaleDenominator>100000</MaxScaleDenominator>
```

Betekent dit dat de regel werkt wanneer de kaart een schaal van 1:50000 tot 1:100000 heeft.

```
<MinScaleDenominator>100000</MinScaleDenominator>
```

Betekent dit dat de regel pas werkt wanneer de kaart een schaal van 1:100000 of meer.

Let op: Er moet voor elke gewenst onderscheid op basis van de schaal een nieuwe regel gemaakt worden. Wanneer punten alleen zichtbaar mogen zijn vanaf een bepaald zoomniveau hoeft er maar één regel te zijn die aan geeft wat de `<MaxScaleDenominator>` is, maar bij veranderingen in de verbeelding naarmate er meer ingezoomd moet worden, moeten er dus meerdere `<rule>`'s komen.

Punten symbolen toevoegen

Zoals eerder al aangegeven zijn er een aantal standaard symbolen in Geoserver welke gebruikt kunnen worden om een punt weer te geven:

- square

- circle
- triangle
- star
- cross
- x

Deze symbolen zitten standaard in Geoserver, maar het is ook mogelijk om andere symbolen te gebruiken hiervoor. Dit kunnen bijvoorbeeld symbolen uit text fonts zijn van het type TTF, maar ook eigen afbeeldingen (.png, .svg, .jpeg).

Symbol uit een .ttf toevoegen

1. Plaats de .ttf direct in de Geoserver data directory:
[Geoserver_data_directory]/styles/[eventuele submap]/bestand.ttf
2. Log in op Geoserver en ga naar 'Server status' onder 'Informatie & status' aan de linkerkant. Klik daar op 'Vrijmaken' achter 'Tijdelijke opslag' en daarna op 'Herladen' achter 'configuratie en catalogus' (zie volgende pagina).

The screenshot shows the Geoserver web interface. The left sidebar contains a menu with 'Server status' highlighted. The main content area displays server status information in a table format. The 'Tijdelijke opslag' row has a 'Vrijmaken' button, and the 'Configuratie en catalogus' row has a 'Herladen' button. Both buttons are highlighted with red boxes. The top right of the page shows 'Ingelogd als admin.' and 'Uitloggen'.

3. Ga naar 'Stijlen' onder 'Configuratie' en klik de gewenste stijl aan.
4. Plaats het volgende tussen <WellKnownName> en </WellKnownName> (onder <Mark> en boven <Fill>)

ttf://[naam van het .tff bestand]#[hexcode van het symbool]

De hexcode is te vinden door het font te bekijken via 'speciale tekens' (in Windows: programma's → accessoires → speciale tekens (wellicht nog in een submap van accessoires). Klik daar het symbool aan en links onderaan staat dan een code in de trant van U+1234, maak daar 0x1234 van.

Bijvoorbeeld: *ttf://Webdings#0x0064*

5. Vul eventueel nog andere eigenschappen van het symbool in (kleur, grootte, transparantie etc.)
6. Klik op 'toepassen'
7. Controleer de stijl door terug te gaan naar de zojuist aangepaste stijl en te klikken op 'Preview legend'.

Een externe afbeelding als symbool gebruiken

Een externe afbeelding kan als symbool gebruikt worden. Een afbeelding wordt dan gebruikt zoals deze al bestaat, er kan bijvoorbeeld geen kleur of rand meegegeven worden.

Als het gaat om een symbool wat op te halen is vanaf een website, gebruik dan het onderstaande stuk. Het is ook mogelijk om de grootte aan te passen door na `</ExternalGraphic>` een `<size>` element toe te voegen.

```
<PointSymbolizer>
  <Graphic>
    <ExternalGraphic>
      <OnlineResource xlink:type="simple"
xlink:href="http://mywebsite.com/pointsymbol.png" />
      <Format>image/[extensie zonder .]</Format>
    </ExternalGraphic>
  </Graphic>
</PointSymbolizer>
```

De bestandsnamen van daadwerkelijke bestanden (.jpeg, .png, .svg) welke gebruikt worden voor het symbool worden door Geoserver automatisch gezien als een relatief ten opzichte van `[Geoserver_data_directory]/styles/`. Dit betekent dat het niet nodig is om de hele zin `[Geoserver_data_directory]/styles/[eventuele submap/]bestandsnaam.png` in te typen, maar dat volstaan kan worden met: `[eventuele submap/]bestandsnaam.png`.

```
<PointSymbolizer>
  <Graphic>
    <ExternalGraphic>
      <OnlineResource xlink:type="simple"
xlink:href="[afbeeldingslocatie vanaf
[Geoserver_data_directory]/styles/]" />
      <Format>image/[extensie zonder .]</Format>
    </ExternalGraphic>
    <Size>Pixels</Size>
  </Graphic>
</PointSymbolizer>
```

Indien het symbool op een andere locatie op de server staat, vul dan de *volledige* locatie van het bestand in.

In het geval van een .svg, vul dan eventueel achter `<Format>` het volgende in: `image/svg+xml`

SLD's voor lijnen

Een simpele lijn SLD (zonder begin en eind van het volledige SLD) ziet er als onderstaande afbeelding uit.

```

1  <FeatureTypeStyle>
2      <Rule>
3          <LineSymbolizer>
4              <Stroke>
5                  <CssParameter name="stroke">#000000</CssParameter>
6                  <CssParameter name="stroke-width">3</CssParameter>
7              </Stroke>
8          </LineSymbolizer>
9      </Rule>
10 </FeatureTypeStyle>

```

Lijnen met een rand

Lijnen hebben geen <Fill>, zoals punten en polygonen. Om een lijn met een rand te maken moeten daarom twee lijnen over elkaar getekend worden.

```

<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">kleur</CssParameter>
        <CssParameter name="stroke-width">pixels</CssParameter>
        <CssParameter name="stroke-linecap">Vorm van het einde
          van de lijn</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">kleur</CssParameter>
        <CssParameter name="stroke-width">pixels</CssParameter>
        <CssParameter name="stroke-linecap">Vorm van het einde
          van de lijn</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>

```

Hier geeft "stroke" de kleur van de lijn weer in hexadecimale kleuren (#FFFFFF bijvoorbeeld). De "stroke-width" geeft in pixels de breedte van de lijn weer. "stroke-linecap" geeft het soort einde van de lijn weer: 'butt',

'round' en 'square' zijn hier de mogelijkheden.

Let hierbij op dat de eerst genoemde <FeatureTypeStyle> als eerste getekend wordt en dus 'onderop' getekend wordt.

Onderbroken lijnen

Stippelijnen kunnen gemaakt worden met de CssParameter "stroke-dasharray". Voeg het volgende toe aan <stroke>:

```
<CssParameter name="stroke-dasharray">[pixels getekende lijn]
[pixels vrije ruimte]</CssParameter>
```

Als bijvoorbeeld het volgende ingevuld staat:

```
<CssParameter name="stroke-dasharray">5 2</CssParameter>
```

Betekent dit dat er 5 getekende pixels zijn, gevolgd door 2 pixels vrije ruimte, gevolgd door 5 getekende pixels etc.

Lijnen met dwarsstreepjes (spoorweg patroon)

Voeg na de </LineSymbolizer> waarmee je de standaard lijn symboliseerd het volgende toe:

```
<LineSymbolizer>
  <Stroke>
    <GraphicStroke>
      <Graphic>
        <Mark>
          <WellKnownName>shape://vertline</WellKnownName>
          <Stroke>
            <CssParameter name="stroke">Kleurcode</CssParameter>
            <CssParameter name="stroke-width">pixels</CssParameter>
          </Stroke>
        </Mark>
        <Size>pixels</Size>
      </Graphic>
    </GraphicStroke>
  </Stroke>
</LineSymbolizer>
```

De "stroke-width" stelt de breedte van de uitstekende streepjes aan beide kanten van de hoofdlijn in op X pixels breed. De <Size> geeft aan om de hoeveel pixels er een dwarsstreepje gezet moet worden.

Lijnen door middel van symbolen

Een lijn aan de hand van symbolen wordt aangegeven met <GraphicStroke>, waarna wordt overgegaan op een stuk <Graphic> zoals dat ook gedaan wordt met punten. In onderstaand voorbeeld wordt er gebruik gemaakt van achtereenvolgende cirkels om een lijn weer te geven


```
<LineSymbolizer>
  <Stroke>
    <GraphicStroke>
      <Graphic>
        <Mark>
          <WellKnownName>circle</WellKnownName>
          <Fill>
            <CssParameter name="fill">kleurcode</CssParameter>
          </Fill>
          <Stroke>
            <CssParameter name="stroke">kleurcode</CssParameter>
            <CssParameter name="stroke-width">pixels</CssParameter>
          </Stroke>
        </Mark>
        <Size>pixels</Size>
      </Graphic>
    </GraphicStroke>
    <CssParameter name="stroke-dasharray">[pixels getekende
lijn] [pixels vrije ruimte]</CssParameter>
  </Stroke>
</LineSymbolizer>
```

Bovenstaande stuk produceert een lijn op basis van cirkels, met een gekleurde vullen, een gekleurde rand van X pixels breed, bepaald hoeveel pixels groot de cirkels mogen zijn en daarna wordt er ingesteld om de hoeveel pixels een cirkeltje getekend wordt. Het is ook mogelijk dit te doen met externe afbeeldingen.

Lijnen met labels

Net als bij punten met labels werkt het geven van labels aan lijnen aan de hand van een <TextSymbolizer>. Het label **kan** met behulp van elementen een stijl meegegeven worden en met behulp van <VendorOption> kan het gedrag van het label bepaald worden. Zie hieronder voor een voorbeeld daarvan:

```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">Kleurcode</CssParameter>
      </Stroke>
    </LineSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>Attribuutnaam</ogc:PropertyName>
      </Label>
      <LabelPlacement>
        <LinePlacement>
          <PerpendicularOffset>10</PerpendicularOffset>
        </LinePlacement>
      </LabelPlacement>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

```
<Fill>
  <CssParameter name="fill">Kleurcode</CssParameter>
</Fill>
<Font>
  <CssParameter name="font-family">Arial</CssParameter>
  <CssParameter name="font-size">10</CssParameter>
  <CssParameter name="font-style">normal</CssParameter>
  <CssParameter name="font-weight">bold</CssParameter>
</Font>
<VendorOption name="followLine">true</VendorOption>
<VendorOption name="maxAngleDelta">90</VendorOption>
<VendorOption name="maxDisplacement">400</VendorOption>
<VendorOption name="repeat">150</VendorOption>
</TextSymbolizer>
</Rule>
</FeatureTypeStyle>
```

In het bovenstaande voorbeeld krijgt een lijn een label aan de hand van een te kiezen attribuut ([attribuutnaam](#)). Het font hierbij is Arial, grootte 10, normale stijl, maar dikgedrukt.

<PerpendicularOffset> binnen *<LabelPlacement>* zorgt er in dit geval voor dat het label 10 pixels boven de lijn zweeft.

Het gedrag van het label is zo ingesteld dat het label de lijn volgt:

```
<VendorOption name="followLine">true</VendorOption>
```

Het label kan een maximale hoek van 90 graden maken (niet meer, om zo een overlap van letters te voorkomen):

```
<VendorOption name="maxAngleDelta">90</VendorOption>
```

Een label kan maximaal 400 pixels verplaatst worden om ruimte te maken voor andere labels, om zo overlap van labels te voorkomen:

```
<VendorOption name="maxDisplacement">400</VendorOption>
```

En het label wordt om de 150 pixels herhaalt:

```
<VendorOption name="repeat">150</VendorOption>
```

SLD's voor polygonen

Een simpele lijn SLD (zonder begin en eind van het volledige SLD) ziet er als onderstaande afbeelding uit.

```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#000080</CssParameter>
      </Fill>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Er staat nu alleen: 'Maak alle vlakken kleur #000080'

Het is ook mogelijk om transparantie toe te voegen om de vlakken doorzichtig te maken. Gebruik daarvoor:

```
<CssParameter name="fill-Opacity">Waarde van 0 tot 1</CssParameter>
```

Polygonen met randen

Om onderscheid te kunnen maken tussen meerdere aaneen grenzende vlakken op een kaart is het handig om randen aan de vlakken toe te voegen. Gebruik daar onderstaande code voor:

```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#Kleurcode</CssParameter>
        <CssParameter name="fill-Opacity">0 tot 1</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">Kleurcode</CssParameter>
        <CssParameter name="stroke-width">Pixels</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Het stuk tussen <Stroke> en </Stroke> geeft een kleur en een breedte in pixels aan de lijn.

Polygonen met gebufferde randen

Het is ook mogelijk om de lijnen van polygoon extra te benadrukken met een buffer. Het stukje `<PerpendicularOffset></PerpendicularOffset>` geeft hier een buffer aan de binnenkant van de polygoon.

```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#Kleurcode</CssParameter>
        <CssParameter name="stroke-width">Pixels</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#Kleurcode</CssParameter>
        <CssParameter name="stroke-width">Pixels</CssParameter>
      </Stroke>
      <PerpendicularOffset>(negatieve)
      Pixels</PerpendicularOffset>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

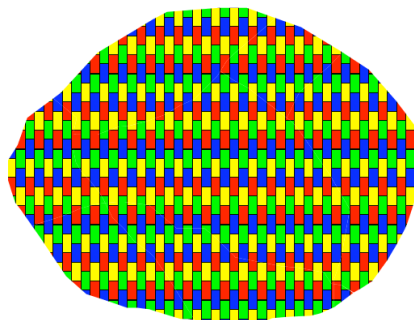
Polygonen gevuld met symbolen

Eerdere voorbeelden gingen uit van een standaard gekleurde vulling. Het is daarnaast ook mogelijk om een polygoon te vullen met een externe afbeelding. De grootte van het plaatje is niet relevant. Als het plaatje te klein zou zijn, wordt het plaatje meerdere keren naast elkaar gebruikt, bijvoorbeeld:

Het originele plaatje:



Na gebruik voor het vullen van een polygoon:



Gebruik hiervoor onderstaande code:

```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <GraphicFill>
          <Graphic>
            <ExternalGraphic>
              <OnlineResource xlink:type="simple"
                xlink:href="http://mywebsite.com/symbol.[png]" />
              <Format>image/[extensie zonder .]</Format>
            </ExternalGraphic>
            <Size>Pixels</Size>
          </Graphic>
        </GraphicFill>
      </Fill>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Polygonen met labels

Het toevoegen van een label aan een polygoon kan met onderstaande code:

```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">Kleurcode</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">Kleurcode</CssParameter>
        <CssParameter name="stroke-width">Pixels</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>name</ogc:PropertyName>
      </Label>
      <Font>
        <CssParameter name="font-family">Arial</CssParameter>
        <CssParameter name="font-size">11</CssParameter>
        <CssParameter name="font-style">normal</CssParameter>
        <CssParameter name="font-weight">bold</CssParameter>
      </Font>
      <LabelPlacement>
        <PointPlacement>
          <AnchorPoint>
            <AnchorPointX>0 tot 1</AnchorPointX>
            <AnchorPointY>0 tot 1</AnchorPointY>
          </AnchorPoint>
        </PointPlacement>
      </LabelPlacement>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

```
</PointPlacement>
</LabelPlacement>
<Fill>
  <CssParameter name="fill">Kleurcode</CssParameter>
</Fill>
  <VendorOption name="autoWrap">Pixels</VendorOption>
  <VendorOption name="maxDisplacement">Pixels</VendorOption>
</TextSymbolizer>
</Rule>
</FeatureTypeStyle>
```

In het bovenstaande voorbeeld wordt er een label gegenereerd door <TextSymbolizer>. Het stukje <Label> geeft aan waar het label vandaan gehaald wordt (welke kolom), geeft een stijl mee aan het lettertype, <LabelPlacement> bepaalt waar het label komt te staan ten opzichte van het midden van de polygoon. Hier wordt een waarde van 0 tot 1 gebruikt, 0.5 is hier het midden. 'autoWrap' bepaalt de maximale breedte van een label op een x aantal pixels, waarna het label doorgaat op een volgende regel. 'maxDisplacement' kan gebruikt worden om te voorkomen dat het label overloopt in een volgend polygoon. Labels worden een x aantal pixels binnen een polygoon verplaatst om het label in zijn geheel binnen een polygoon te plaatsen.

Polygonen met stijlen op basis van attributen

Wanneer u een kaart wilt maken waarbij polygonen verbeeld worden op basis van het voldoen aan een bepaalde voorwaarde, moet u meerdere malen een <rule> invoegen. Elke <rule> definieert dan hoe een polygoon verbeeld moet worden wanneer aan een voorwaarde voldaan wordt.

Onderstaande code is daar een voorbeeld van:

```
<FeatureTypeStyle>
  <Rule>
    <Name>Name</Name>
    <Title>Title</Title>
    <ogc:Filter>
      <ogc:PropertyIsLessThan>
        <ogc:PropertyName>Attribuut</ogc:PropertyName>
        <ogc:Literal>Waarde</ogc:Literal>
      </ogc:PropertyIsLessThan>
    </ogc:Filter>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">Kleurcode</CssParameter>
      </Fill>
    </PolygonSymbolizer>
  </Rule>
  <Rule>
    <Name>Name</Name>
    <Title>Title</Title>
    <ogc:Filter>
      <ogc:And>
```

```
<ogc:PropertyIsGreaterThanOrEqualTo>
  <ogc:PropertyName>Attribuut</ogc:PropertyName>
  <ogc:Literal>Waarde</ogc:Literal>
</ogc:PropertyIsGreaterThanOrEqualTo>
<ogc:PropertyIsLessThan>
  <ogc:PropertyName>Attribuut</ogc:PropertyName>
  <ogc:Literal>Waarde</ogc:Literal>
</ogc:PropertyIsLessThan>
</ogc:And>
</ogc:Filter>
<PolygonSymbolizer>
  <Fill>
    <CssParameter name="fill">#33CC33</CssParameter>
  </Fill>
</PolygonSymbolizer>
</Rule>
<Rule>
  <Name>Name</Name>
  <Title>Title</Title>
  <ogc:Filter>
    <ogc:PropertyIsGreaterThanOrEqualTo>
      <ogc:PropertyName>Attribuut</ogc:PropertyName>
      <ogc:Literal>Waarde</ogc:Literal>
    </ogc:PropertyIsGreaterThanOrEqualTo>
  </ogc:Filter>
  <PolygonSymbolizer>
    <Fill>
      <CssParameter name="fill">Kleurcode</CssParameter>
    </Fill>
  </PolygonSymbolizer>
</Rule>
</FeatureTypeStyle>
```

In bovenstaande voorbeeld zijn er 3 verschillende soorten polygoenen, de stukken <ogc:filter> bepalen welke objecten er meegenomen worden bij gebruiken van een bepaalde <rule> bij het genereren van een kaart.

De eerste <rule> bepaald dat alle objecten die een attribuut hebben met een waarde kleiner dan X, die stijl moeten krijgen. De tweede <rule> bepaald dat alle objecten die een attribuut hebben met een waarde groter dan X en kleiner dan Y, die stijl krijgen (let hier ook op <ogc:and>, dit zorgt ervoor dat aan beide voorwaarden voldaan moet worden). De derde <rule> bepaald dan dat objecten die voldoen aan een waarde die groter is dan Y die stijl moeten krijgen.

Polygonen met stijlen op basis van zoomniveau's

Soms is het wenselijk om bepaalde stijlen pas te laten zien vanaf een bepaald zoomniveau, bijvoorbeeld labels. Hiervoor geldt ook weer dat voor ieder zoomniveau een nieuwe <rule> gemaakt moet worden

```
<FeatureTypeStyle>
  <Rule>
    <Name>Name</Name>
    <MaxScaleDenominator>Waarde</MaxScaleDenominator>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">Kleurcode</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">Kleurcode</CssParameter>
        <CssParameter name="stroke-width">Pixels</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>Attribuut</ogc:PropertyName>
      </Label>
      <Font>
        <CssParameter name="font-family">Arial</CssParameter>
        <CssParameter name="font-size">14</CssParameter>
        <CssParameter name="font-style">normal</CssParameter>
        <CssParameter name="font-weight">bold</CssParameter>
      </Font>
      <LabelPlacement>
        <PointPlacement>
          <AnchorPoint>
            <AnchorPointX>0.5</AnchorPointX>
            <AnchorPointY>0.5</AnchorPointY>
          </AnchorPoint>
        </PointPlacement>
      </LabelPlacement>
      <Fill>
        <CssParameter name="fill">Kleurcode</CssParameter>
      </Fill>
    </TextSymbolizer>
  </Rule>
  <Rule>
    <Name>Medium</Name>
    <MinScaleDenominator>Waarde</MinScaleDenominator>
    <MaxScaleDenominator>Waarde</MaxScaleDenominator>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">Kleurcode</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">Kleurcode</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```



```
        <CssParameter name="stroke-width">Pixels</CssParameter>
    </Stroke>
</PolygonSymbolizer>
</Rule>
</FeatureTypeStyle>
```

De verschillende zoomniveau's worden ingesteld met onderstaande elementen:

```
<MinScaleDenominator></MinScaleDenominator>
<MaxScaleDenominator></MaxScaleDenominator>
```

Bovenstaande code zorgt ervoor dat wanneer een eindgebruiker ver inzoomt, er een label gegenereerd wordt bij het polygoon. Verder uitzoomen verwijderd dat label weer uit het beeld.