Flexmap (technische) Beschrijving

geo

Openwave levert standaard de flexmap schermen uit. Hieronder volgt een uitleg hoe deze schermen door functioneel beheerders aangepast kunnen worden naar eigen behoefte : wat is op de kaart te zien en met welke bewoordingen. Ook wordt beschreven met welke functie-aanroep de flexmap geopend kan worden, zowel in kijkmodus als in tekenmodus. De userinterafce van de kaart is beschreven in Flexmap knoppen en widgets

Gebruik en aanroep

De (interne) FlexMap wordt getoond indien

- de instelling *Sectie: Programma en Item: Flexmap* aangevinkt is.
- EN de instelling *Sectie: Programma en Item: ToonKaart* aangevinkt is
- EN de kolom info van de instelling *Sectie: Programma en Item: Toonkaart* leeg is (hier kan namelijk de URL van een externe kaartviewer staan).

Is dat het geval dan wordt met de kaartknop of met een tekenopdracht uit een menu-lijstje de flexmap getoond door een aanroep naar de OpenWave-api *getFlexMap*. Daarbij wordt met parameters aangegeven of de kaart alleen in kijkmodus wordt geopend, of dat de kaart geopend wordt om een punt aan te wijzen of om een polygoon of lijn te tekenen. Ook wordt met een parameter geregeld waarvandaan de kaart wordt geopend en wat er op te zien is, door de naam van een xml-file mee te geven, waarin deze informatie staat.

let op

De aanroep voor de nieuwe kaart wordt niet standaard uitgeleverd in het openingsportaal: men dient deze zelf aan te maken. Gebruik hiervoor de volgende **action**: *getFlexMap(1,tbportalnames,MDMC_AlleOpenstaandeZaken.xml)*. Wanneer bovengenoemde instellingen aanstaan, zal de nieuwe kaart aangeroepen kunnen worden.

Andere voorbeelden aanroep

- vanuit een omgevingzaak in kijkmodus: *getFlexMap(163636,tbomgvergunning,MDMC_tbomgvergunning.xml)* waarbij 163636 de dnkey is van een kaart in tbomgvergunning
    - dus als action bij een zelfgedefinieerde knop op een detailschem *getFlexMap(%keypointer%,tbomgvergunning,MDMC_tbomgvergunning.xml)*
    - dus als action bij een zelfgedefinieerde knop op een lijstscherm *getFlexMap({id},tbomgvergunning,MDMC_tbomgvergunning.xml)*
- vanuit een inrichting met de opdracht om een vlak te tekenen: *getFlexMap(13456,tbmilinrichtingen,MDMC_tbmilinrichtingen.xml,Polygon)*
- vanuit een projectlocatie die gekoppeld is aan een omgevingzaak met de opdracht om een punt aan te wijzen: *getFlexMap(12345,tbzaakkadperc,MDMC_tbzaakkadperc_w.xml,Point)*
- vanuit een projectlocatie die gekoppeld is aan een inrichting in kijkmodus:

getFlexMap(6789,tbzaakkadperc,MDMC_tbzaakkadperc_V.xml)
- vanuit een stal met de opdracht om een vlak te tekenen
  getFlexMap(2345,tbmilstal,MDMC_tbmilstal.xml,Polygon)

De api getFlexmap heeft dus 3 of 4 parameters:

- param1 = een pointer naar een specifieke rij van een tabel van waaruit de flexmap wordt aangeroepen door middel van de primary key van die tabel.
- param2 = de tabelnaam van waaruit de flexmap wordt aangeroepen
- param3 = de naam van de xml waarin alle instructies staan voor de opbouw van de kaart. De naam van deze xml moet beginnen met '*MDMC_*'
- param4 = indien
  - leeg dan wordt de kaart gestart in kijkmodus.
  - Indien *Polygon* dan kan de gebruiker een polygoon tekenen op de kaart.
  - Indien *Point* dan kan de gebruiker een punt aanwijzen op de kaart
  - Indien *LineString* dan kan de gebruiker een lijn trekken op de kaart

Welke MDMC_*.xml's

In de aangeroepen MDMC_*.xml is geregeld:

- het centrale punt van de kaart
- wat er in de widget info getoond moet worden
- welke basisobjecten er sowieso zichtbaar moeten zijn
- wat er in de widget tekenopdracht getoond moet worden
- welke kaartlagen er getoond kunnen worden
- waar en hoe een getekend punt of polygoon of lijn moet worden opgeslagen

In de programmatuur van OpenWave zijn hiertoe standaard de volgende xml's opgenomen

- MDMC_AlleOpenstaandeZaken.xml
- MDMC_tbperceeladressen
- MDMC_tbmilinrichtingen
- MDMC_tbomgvergunning
- MDMC_tbomgvergunning
- MDMC_tbmilvergunningen
- MDMC_tbhandhavingen
- MDMC_tbinfoaanvragen
- MDMC_tbhorecavergunningen
- MDMC_tbbouwvergunningen
- MDMC_tbzaakkadperc_w (voor een project locatie gekoppeld aan een omgevingzaak)
- MDMC_tbzaakkadperc_o (voor een project locatie gekoppeld aan een apv/overige zaak)
- MDMC_tbzaakkadperc_h (voor een project locatie gekoppeld aan een handhaving zaak)
- MDMC_tbzaakkadperc_v voor een project locatie gekoppeld aan een inrichting)
- MDMC_tbmilopslag
- MDMC_tbmildiversen
- MDMC_tbmilemlucht
- MDMC_tbmilemwater
- MDMC_horontheffingen

- MDMC_tbmilstal
- MDMC_tbmilasbest
- MDMC_tbmilopslagevcontour
- MDMC_tbmilbklkwetsbgebloc.xml

OpenWave zoekt bij het openen van de kaart eerst naar de waarde van param3 van de Flexmap-aanroep in de tabel tbscreencolumns in de kolom dvscreenfilename (*beheerportaal: tegel Alle schermkolomdefinities*) en als de kolom dvscreenxml gevuld is, dan wordt die (xml-)definitie genomen. Niet gevonden, of de kolom dvscreenxml is leeg, dan gebruikt OpenWave de default xml-kaartdefinitie, maar dit alleen indien param3 één van de bovenstaande namen heeft.

Foutafvanging

Indien de kaart wordt aangeroepen op basis van een xml die niet goed in elkaar zit, dan

- verschijnt een foutmelding linksonder in het scherm
- en wordt een rij aangemaakt in de tabel tbmissingconfiguration (beheerportaal: tegel: ontbrekende configuratieitems) met in de foutbeschrijving het gedeelte uit de xml dat voor het probleem zorgt.

De xml moet in ieder geval aan de volgende xsd voldoen

```
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
        <xs:element name="document">
                <xs:complexType>
                        <xs:sequence>
                                 <xs:element ref="center" minOccurs="1"
maxOccurs="1"/>
                                <xs:element name="zoom" type="xs:integer"
minOccurs="1" maxOccurs="1"/>
                                <xs:element ref="widgetinfo" minOccurs="1"
maxOccurs="1"/>
                                <xs:element ref="basisobjecten"
minOccurs="0" maxOccurs="1"/>
                                <xs:element ref="widgetopdracht"
minOccurs="0" maxOccurs="1"/>
                                <xs:element ref="openwavelayers"
minOccurs="0" maxOccurs="1"/>
                                <xs:element ref="wfslayers" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="wmslayers" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="widgetbuttons"
minOccurs="0" maxOccurs="1"/>
                                <xs:element ref="actions" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="punten" minOccurs="0"
maxOccurs="1"/>
```

```
                                <xs:element ref="lijnen" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="vlakken" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="cirkels" minOccurs="0"
maxOccurs="1"/>
                          </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="center">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="point" type="xs:string"
minOccurs="1" maxOccurs="1"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="widgetinfo">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="koptekst" minOccurs="1"
maxOccurs="1"/>
                                <xs:element ref="body" minOccurs="1"
maxOccurs="1"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="koptekst">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="regel"
type="koptekstregeltype" minOccurs="1" maxOccurs="unbounded"/>
                         </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:complexType name="koptekstregeltype">
                <xs:simpleContent>
                        <xs:extension base="xs:string">
                                <xs:attribute name="id" use="required"
type="xs:integer"/>
                         </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
        <xs:element name="body">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="regel"
type="koptekstregeltype" minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
```

```
        <xs:element name="basisobjecten">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="basisobject" minOccurs="1"
maxOccurs="unbounded"/>
                         </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="basisobject">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="resultset"
type="xs:string" minOccurs="1" maxOccurs="1"/>
                        </xs:sequence>
                        <xs:attribute name="type" use="required"
type="xs:string"/>
                </xs:complexType>
        </xs:element>
        <xs:element name="widgetopdracht">
                <xs:complexType>
                        <xs:all>
                                <xs:element ref="point" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="linestring" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="polygon" minOccurs="0"
maxOccurs="1"/>
                         </xs:all>
                </xs:complexType>
        </xs:element>
        <xs:element name="polygon">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="koptekst"  minOccurs="1"
maxOccurs="1"/>
                                <xs:element ref="opdracht"  minOccurs="1"
maxOccurs="1"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="opdracht">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="resultset"
type="xs:string" minOccurs="1" maxOccurs="1"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="point">
                <xs:complexType>
                        <xs:sequence>
```

```
                                <xs:element ref="koptekst"  minOccurs="1"
maxOccurs="1"/>
                                <xs:element ref="opdracht"  minOccurs="1"
maxOccurs="1"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="linestring">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="koptekst"  minOccurs="1"
maxOccurs="1"/>
                                <xs:element ref="opdracht"  minOccurs="1"
maxOccurs="1"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="openwavelayers">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="layer" type="owlayertype"
minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:complexType name="owlayertype">
                <xs:simpleContent>
                        <xs:extension base="xs:string">
                                <xs:attribute name="id" use="required"
type="xs:integer"/>
                                <xs:attribute name="active" use="required"
type="xs:boolean"/>
                        </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
        <xs:element name="wfslayers">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="layer" type="wfslayertype"
minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:complexType name="wfslayertype">
                <xs:simpleContent>
                        <xs:extension base="xs:string">
                                <xs:attribute name="id" use="required"
type="xs:integer"/>
                                <xs:attribute name="active" use="required"
type="xs:boolean"/>
                                <xs:attribute name="isbasislaag"
```

```
use="required" type="xs:boolean"/>
                                <xs:attribute name="opacity" use="required"
type="xs:integer"/>
                                <xs:attribute name="index" use="required"
type="xs:integer"/>
                                <xs:attribute name="layer" use="required"
type="xs:string"/>
                                <xs:attribute name="source" use="required"
type="xs:string"/>
                                <xs:attribute name="color" use="required"
type="xs:string"/>
                                <xs:attribute name="icon" use="required"
type="xs:string"/>
                        </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
        <xs:element name="wmslayers">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="layer" type="wmslayertype"
minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:complexType name="wmslayertype">
                <xs:simpleContent>
                        <xs:extension base="xs:string">
                                <xs:attribute name="id" use="required"
type="xs:integer"/>
                                <xs:attribute name="active" use="required"
type="xs:boolean"/>
                                <xs:attribute name="isbasislaag"
use="required" type="xs:boolean"/>
                                <xs:attribute name="opacity" use="required"
type="xs:integer"/>
                                <xs:attribute name="index" use="required"
type="xs:integer"/>
                                <xs:attribute name="layer" use="required"
type="xs:string"/>
                                <xs:attribute name="source" use="required"
type="xs:string"/>
                        </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
        <xs:element name="widgetbuttons">
                <xs:complexType>
                        <xs:all>
                                <xs:element ref="opdrachtpoint"
minOccurs="0" maxOccurs="1"/>
                                <xs:element ref="opdrachtlinestring"
minOccurs="0" maxOccurs="1"/>
```

```
                                <xs:element ref="opdrachtpolygon"
minOccurs="0" maxOccurs="1"/>
                          </xs:all>
                </xs:complexType>
        </xs:element>
        <xs:element name="opdrachtpoint">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="button" type="buttontype"
minOccurs="1" maxOccurs="unbounded"/>
                         </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:complexType name="buttontype">
                <xs:simpleContent>
                        <xs:extension base="xs:boolean">
                                <xs:attribute name="id" use="required"
type="xs:integer"/>
                                <xs:attribute name="hint" use="required"
type="xs:string"/>
                                <xs:attribute name="icoon" use="required"
type="xs:integer"/>
                                <xs:attribute name="action" use="required"
type="xs:string"/>
                        </xs:extension>
                 </xs:simpleContent>
        </xs:complexType>
        <xs:element name="opdrachtlinestring">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="button" type="buttontype"
minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="opdrachtpolygon">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="button" type="buttontype"
minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="actions">
                <xs:complexType>
                        <xs:all>
                                <xs:element ref="actionpoint" minOccurs="0"
maxOccurs="1"/>
                                <xs:element ref="actionpolygon"
minOccurs="0" maxOccurs="1"/>
                                <xs:element ref="actionlinestring"
```

```
minOccurs="0" maxOccurs="1"/>
                        </xs:all>
                </xs:complexType>
        </xs:element>
        <xs:element name="actionpoint">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="action" type="actiontype"
minOccurs="2" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="actionlinestring">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="action" type="actiontype"
minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="actionpolygon">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="action" type="actiontype"
minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:complexType name="actiontype">
                <xs:simpleContent>
                        <xs:extension base="xs:string">
                                <xs:attribute name="id" use="required"
type="xs:integer"/>
                           </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
        <xs:element name="punten">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="punt" minOccurs="1"
maxOccurs="unbounded"/>
                         </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="punt">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="resultset"
type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="layer" use="required"
```

```
type="xs:integer"/>
                </xs:complexType>
        </xs:element>
        <xs:element name="lijnen">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="lijn" minOccurs="1"
maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="lijn">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="resultset"
type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="layer" use="required"
type="xs:integer"/>
                </xs:complexType>
        </xs:element>
        <xs:element name="vlakken">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="vlak" minOccurs="1"
maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="vlak">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="resultset"
type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="layer" use="required"
type="xs:integer"/>
                </xs:complexType>
        </xs:element>
        <xs:element name="cirkels">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="cirkel" minOccurs="1"
maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="cirkel">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="resultset"
```

```
type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="layer" use="required"
type="xs:integer"/>
                 </xs:complexType>
        </xs:element>
   </xs:schema>
```

Inhoudelijke beschrijving van de MDMC_*.xml

(Op basis van mdmc_tbmilinrichtingen.xml)

De xml is verdeeld in de volgende blokken:

- center
- zoom
- widgetinfo
- basisobjecten
- widgetopdracht
- openwavelayers
- wfslayers
- wmslayers
- widgetbuttons
- actions
- punten
- lijnen
- vlakken
- cirkels

OpenWave evalueert voordat de kaart geopend wordt de sql-statements in de xml en vervangt deze statements door hun resultaat. De variabele %keypointer% wordt overal vervangen door param1 van de flexmap -aanroep (dus in onderstaande voorbeelden een dnkey van tbmilinrichhtingen)

Blok

voorbeeld center

```
<center>
         <point>select case when b.dvxrdcoordinaat is null and
substr(b.dvgmlrdpolygoon, 1, position(' 'IN b.dvgmlrdpolygoon) - 1) is null
then coalesce(a.dnxcoordinaat,155600) ||','||
coalesce(a.dnycoordinaat,465175)
             when b.dvxrdcoordinaat is null then substr(b.dvgmlrdpolygoon,
1, position(' 'IN b.dvgmlrdpolygoon) - 1)
             else b.dvxrdcoordinaat ||','|| b.dvyrdcoordinaat
             end dvpos
             from vwfrmmilinrichtingen a left outer join tbzaakkadperc b
```

```
              on (a.dnkeymilinrichtingen = b.dnkeymilinrichtingen and
b.dlhoofdprojectlocatie = 'T')
              where a.dnkeymilinrichtingen = %keypointer%
          </point>
  </center>
```

Het blok moet één keer voorkomen. Het blok bevat één tag <point> dat gevuld wordt door de evaluatie van het sql-statement. Het resultaat van de sql bestaat uit één rij bestaande uit één string die een x-coordinaat en een y-coordinaat gescheiden door een komma bevat. In rijksdriehoek. De kaart wordt gecentreerd op dat punt

Tag <zoom>

voorbeeld zoom

```
<zoom>12</zoom>
```

De tag <zoom> moet één keer voorkomen. In de tag <zoom> moet rechtsreeks een integerwaarde staan (dus geen sql-statement). De kaart wordt gestart met dit zoomlevel.

Blok <widgetinfo>

voorbeeld widgetinfo

```
<widgetinfo>
        <koptekst>
            <regel id="1">select 'Locatiedossier:' || dvinrichtingnaam from
vwfrmmilinrichtingen where
                dnkeymilinrichtingen = %keypointer%</regel>
            <regel id="2">select coalesce(dvobjadres,'') ||' ' ||
coalesce(dvobjplaats,'') from vwfrmmilinrichtingen
                where dnkeymilinrichtingen = %keypointer%</regel>
        </koptekst>
        <body>
            <regel id="1"> select 'Handelsnaam:' || dvhandelsnaam from
vwfrmmilinrichtingen where
                dnkeymilinrichtingen = %keypointer%</regel>
        </body>
    </widgetinfo>
```

Het blok <widgetinfo> moet één keer voorkomen. Het blok <widgetinfo> moet één blok <koptekst> bevatten en elk blok <koptekst> moet bestaan uit één of meer regels die in het attribuut id oplopend genummerd moeten worden. Het sql-statement per regel moet één rij opleveren bestaande uit één string. Het blok widgetinfo moet ook één blok <body> bevatten en elk blok <body>

moet bestaan uit één of meer regels die in het attribuut id oplopend genummerd moeten worden. Het sql-statement per regel moet één rij opleveren bestaande uit één string. Op de kaart verschijnt een widget met als titel *Zaakinformatie* met daarin de opgegeven kop- en bodyteksten.

Blok <basisobjecten>

voorbeeld basisobjecten

```
<basisobjecten>
        <basisobject type="point">
            <!--kolommen dvcolorname, dvdescription en dvpos zijn verplicht
-->
            <resultset>select 'Adrespunt: '||
coalesce(a.dvinrichtingnaam,'') dvdescription,
                case when b.dvxrdcoordinaat is null then a.dnxcoordinaat
||','|| a.dnycoordinaat
                else b.dvxrdcoordinaat ||','|| b.dvyrdcoordinaat
                end dvpos, 'Red' dvcolorname
                from vwfrmmilinrichtingen a left outer join tbzaakkadperc b
                on (a.dnkeymilinrichtingen = b.dnkeymilinrichtingen and
b.dlhoofdprojectlocatie = 'T')
                where a.dnkeymilinrichtingen = %keypointer%
            </resultset>
        </basisobject>
        <basisobject type="polygon">
            <!--kolommen dvfillcolorname, dvstrokecolorname, dvdescription
en dvposlist en dfopacity zijn verplicht -->
            <resultset>select 'Grondvlak: '||
coalesce(a.dvinrichtingnaam,'') dvdescription,
                coalesce(a.dvgmlpolygoon,b.dvgmlrdpolygoon) dvposlist, 'Red'
                dvfillcolorname, 'Black' dvstrokecolorname, 0.5 dfopacity
from vwfrmmilinrichtingen a left outer join tbzaakkadperc b
                on (a.dnkeymilinrichtingen = b.dnkeymilinrichtingen and
b.dlhoofdprojectlocatie = 'T') where
                a.dnkeymilinrichtingen = %keypointer% and
coalesce(b.dvgmlrdpolygoon,a.dvgmlpolygoon) is not null and
fn_islinestring(coalesce(b.dvgmlrdpolygoon,a.dvgmlpolygoon)) !=
1</resultset>
        </basisobject>
        <basisobject type="linestring">
            <!--kolommen  dvstrokecolorname, dvdescription en dvposlist en
dfopacity zijn verplicht -->
            <resultset>select 'Grondvlak: '||
coalesce(a.dvinrichtingnaam,'') dvdescription,
                coalesce(a.dvgmlpolygoon,b.dvgmlrdpolygoon) dvposlist, 'Red'
                dvfillcolorname, 'Black' dvstrokecolorname, 0.5 dfopacity, 3
dfstrokewidth from vwfrmmilinrichtingen a left outer join tbzaakkadperc b
                on (a.dnkeymilinrichtingen = b.dnkeymilinrichtingen and
b.dlhoofdprojectlocatie = 'T') where
```

```
                a.dnkeymilinrichtingen = %keypointer% and
coalesce(b.dvgmlrdpolygoon,a.dvgmlpolygoon) is not null and
fn_islinestring(coalesce(b.dvgmlrdpolygoon,a.dvgmlpolygoon)) = 1
</resultset>
        </basisobject>
    </basisobjecten>
```

Het blok `<basisobjecten>` mag één keer vookomen (maar hoeft niet). Binnen het blok `<basisobjecten>` moet één of meer keer een blok `<basisobject>` zijn gedefinieerd. Een blok `<basisobject>` heeft het attribuut *type* dat mag bestaan uit *point* of *polygon* of *linestring*

Een blok `<basisobject>` van type *point* bevat één tag `<resultset>` waarachter een sql-statement die na evaluatie één rij oplevert met de kolommen dvcolorname, dvdescription en dvpos waarbij dvpos gevuld moet zijn met een x-coordinaat en een y-coordinaat, in rijksdriehoek, gescheiden door een komma en dvcolorname met een *html-color-name*.

Een blok `<basisobject>` van type *polygon* bevat één tag `<resultset>` waarachter een sql-statement die na evaluatie één rij oplevert met de kolommen dvfillcolorname, dvstrokecolorname, dvdescription en dvposlist en dfopacity waarbij:

- dvfillcolorname (vulling) en dvstrokecolorname (omlijning) gevuld moeten zijn met een *html-color-name*
  * dvposlist met een reeks rijksdriehoek-punten gescheiden door een spatie waarbij het laatste coordinatenpaar gelijk is aan het eerste paar (bijv: 154039.62,464749.74000000005 154154.7,464785.86000000004 154197.54,464748.9000000001 154137.9,464712.7800000001 154092.54,464723.70000000007 154039.62,464749.74000000005)
- dfopacity met een float-waarde tussen 0 en 1

Een blok `<basisobject>` van type *linestring* bevat één tag `<resultset>` waarachter een sql-statement die na evaluatie één rij oplevert met de kolommen dvfillcolorname, dvstrokecolorname, dvdescription en dvposlist en dfopacity waarbij:

- dvstrokecolorname gevuld moet zijn met een *html-color-name*
  * dvposlist met een reeks rijksdriehoek-punten gescheiden door een spatie waarbij het laatste coordinatenpaar ongelijk is aan het eerste paar (bijv: 154039.62,464749.74000000005 154154.7,464785.86000000004 154197.54,464748.9000000001 154137.9,464712.7800000001 154092.54,464723.70000000007)
- dfopacity met een float-waarde tussen 0 en 1

De uitkomst van de sql-statements mag null zijn. In dat geval wordt het betreffende vlak, punt of lijn niet geprojecteerd op de map. De sql-statements die wel valide punt(en) en of vlak(ken) en of lijn(en) retourneren worden geprojecteerd op de kaart en daarvan zijn de omschrijvingen (dvdescription) terug te vinden in de widget met de titel// Legenda//

Blok <widgetopdracht>

voorbeeld widgetopdracht

```
<widgetopdracht>
        <linestring>
            <koptekst>
                <regel id="1">select case when dvgmlpolygoon is null then
'Teken nieuwe lijn' else 'Wijzig lijn' end from vwfrmmilinrichtingen where
                    dnkeymilinrichtingen = %keypointer%</regel>
            </koptekst>
            <opdracht>
                <!--kolommen dvstrokecolorname, dvdescription en dvposlist
zijn verplicht -->
                <resultset>select 'Lijn:' dvdescription,
                    dvgmlpolygoon dvposlist, 'Fuchsia' dvstrokecolorname,
0.5
                    dfopacity, 3 dfstrokewidthfrom vwfrmmilinrichtingen
where dnkeymilinrichtingen = %keypointer% and fn_islinestring(dvgmlpolygoon)
= 1</resultset>
            </opdracht>
        </linestring>
        <polygon>
            <koptekst>
                <regel id="1">select case when dvgmlpolygoon is null then
'Teken nieuw inrichtingcontour' else 'Wijzig inrichtingcontour' end from
vwfrmmilinrichtingen where
                    dnkeymilinrichtingen = %keypointer%</regel>
            </koptekst>
            <opdracht>
                <!--kolommen dvcolorname, dvdescription en dvpos zijn
verplicht -->
                <resultset>select 'Inrichtingscontour:' || dvinrichtingnaam
dvdescription,
                    dvgmlpolygoon dvposlist, 'Fuchsia' dvfillcolorname,
'Black' dvstrokecolorname, 0.5
                    dfopacity from vwfrmmilinrichtingen where
dnkeymilinrichtingen = %keypointer%</resultset>
            </opdracht>
        </polygon>
   </widgetopdracht>
```

Het blok <widgetopdracht> mag één keer vookomen (maar hoeft niet). Indien het blok niet voorkomt is er geen extra informatie over de tekenopdracht (zie hieronder blok <koptekst>) beschikbaar, wanneer de vierde parameter van de Flexmap-aanroep is gevuld. Binnen het blok <widgetopdracht> mag één keer een blok <point> en mag één keer een blok <linestring> en mag één keer een blok <polygon> zijn gedefinieerd. Binnen een blok <widgetopdracht><point> of <widgetopdracht><polygon> of <widgetopdracht><linestring> moet

- één blok <koptekst> zijn gedefinieerd met daarin één of meer tags <regel> die in het attribuut id oplopend genummerd moeten worden. Achter een tag regel moet een sqlstatement staan dat na evaluatie één rij bestaande uit één string oplevert.
- en één blok <opdracht> met daarbinnen één tag <resultset> gevuld met een sql-statement dat na evaluatie:

- in geval van <point> één rij met de kolommen dvcolorname, dvdescription en dvpos bevat (zie hierboven bij <basisobject type="point">)
- in geval van <polygon> één rij met de kolommen dvfillcolorname, dvstrokecolorname, dvdescription en dvposlist en dfopacity bevat (zie hierboven bij <basisobject type = "polygon">)
- in geval van <linestring> één rij met de kolommen dvstrokecolorname, dvdescription en dvposlist en dfopacity bevat (zie hierboven bij <basisobject type="linestring">)

De evaluatie van de tag `<resultset>` mag null zijn. Dus *select null* is een geldig statement. Met deze informatie wordt in deze Openwave versie (namelijk nog) niets gedaan.

De koptekst wordt gebruikt als tekst in de widget met de titel *tekenen en bewerken* die zichtbaar wordt indien de aanroep van de Flexmap in de vierde parameter de waarde *Polygon* of *Point* of *LineString* bevat (indien point dan wordt de koptekst van <widgetopdracht><point> gebruikt, indien linestring dan die van <widgetopdracht><linestring> en bij Polygon die van <widgetopdracht><polygon>.

Blok <openwavelayers>

voorbeeld openwavelayers

```
<openwavelayers>
        <layer id="3" active="false">select 'REV Referentieobjecten'</layer>
        <layer id="4" active="false">select 'REV EvContouren'</layer>
        <layer id="5" active="false">select 'Stallen'</layer>
        <layer id="6" active="false">select 'Projectlocaties'</layer>
        <layer id="7" active="false">select 'Perceeladrespunt bij
inrichting'</layer>
</openwavelayers>
```

Het blok `<openwavelayers>` mag één keer vookomen (maar hoeft niet). het blok `<openwavelayers>` bestaat uit een of meer tags <layer>. Elke tag `<layer>` heeft een attribuut *id* (elke layer heeft hierin een unieke integer waarde) waarnaar wordt verwezen in de blokken <vlakken>, <punten>, <lijnen> en cirkels (zie verderop). Elke tag `<layer>` heeft een attribuut *active* die de waarde false of true kan hebben: de layer staat standaard aan of uit. Achter de tag `<layer>` staat een sql-statemnet dat na evaluatie één rij oplevert bestaande uit één string: de laagnaam. Deze laagnamen zijn aan en uit te vinken op de kaart in de widget met de titel *filter & kaartlagen* onder het subkopje *data-objecten*. Indien een laagnaam is aangevinkt dan zal Openwave de punten, vlakken, lijnen en cirkels tonen van de blokken vlak, lijn, punt en cirkel met een overeenkomstig attribuut layer id (zie verderop). Deze getoonde openwave-lagen bestaan uit data die komen uit de OpenWave database.

let op

De openwavelayers zijn bedoeld om een beperkt aantal punten, lijnen of vlakken te tonen. Waneer het bijvoorbeeld gaat om alle zaken van een ruime periode te tonen kan beter gebruik gemaakt worden van wfs-kaartlagen of wms-kaartlagen. Zie: Geo-WMS/WFS lagen

Blok <wfslayers>

voorbeeld wfslayers

```
<wfslayers>
      <layer active="false" id="5" index="5" isbasislaag="false"
layer="ad:address"
             opacity="50"
source="https://service.pdok.nl/kadaster/ad/wfs/v1_0?" icon="2"
             color="Aquamarine">adressen</layer>
</wfslayers>
```

Het blok `<wfslayers>` mag één keer vookomen (maar hoeft niet). het blok `<wfslayers>` bestaat uit een of meer tags <layer>. Elke tag `<layer>` heeft een attribuut:

- id (elke layer heeft hierin een unieke integer waarde).
- active die de waarde false of true kan hebben: de wfslayer staat standaard aan of uit.
- index die een integerwaarde heeft
- isbasislaag die altijd de waarde false moet hebben
- layer met de externe naam van de wfs-laag
- opacity met een integerwaarde tussen 0 en 100
- source met de url waar de laag is op te halen
- icon met een integerwaarde 1, 2 of 3 die aangeeft met welk soort icoontje de punten op de wfslaag getoond moeten worden (respectievelijk vierkant, rondje, driehoek)
- color met een html_color-name

Achter de tag `<layer>` staat een string (dus geen sql-stament) met de naam zoals de laag op de kaart moet heten: de laagnaam. Deze laagnamen zijn aan en uit te vinken op de kaart in de widget met de titel *filter & kaartlagen* onder het subkopje WFS-lagen. Indien een laagnaam is aangevinkt dan zal Openwave de betreffende externe laag ophalen en tonen.

LET OP:

De WFS-lagen die in de tabel tbgeowms (*beheerportaal tegel: Geo kaartlagen*) aangemerkt zijn

- als WFS-laag: de kolom dlwMS *WMS-laag? (anders WFS)* staat NIET aangevinkt
- EN waarvoor geldt dat de kolom dlinFlexmap (*Altijd opnemen in Flexmap*) wel aangevinkt is

die lagen worden onder water automatisch opgenomen in de xml ook als het blok <wfslayers> niet bestaat.

Blok <wmslayers>

voorbeeld wmslayers

```
<wmslayers>
```

```
       <layer active="false" id="1" index="1" isbasislaag="false"
layer="pand" opacity="50"
source="https://service.pdok.nl/lv/bag/wms/v2_0">BAG-panden</layer>
       <layer active="false" id="2" index="2" isbasislaag="false"
layer="Kadastralekaart" opacity="50"
source="https://service.pdok.nl/kadaster/kadastralekaart/wms/v5_0">Kadaster<
/layer>
       <layer active="true" id="3" index="3" isbasislaag="false"
layer="OpenWave:GeotestVanAnnelies" opacity="70"
source="/geoserver/OpenWave/wms">GeotestVanAnnelies</layer>
       <layer active="false" id="4" index="4" isbasislaag="false"
layer="OpenWave:GEO_Actuele_vergunning" opacity="100"
source="/geoserver/OpenWave/wms?">Alle vergunningszaken</layer>
</wmslayers>
```

Het blok `<wmslayers>` mag één keer vookomen (maar hoeft niet). het blok `<wmslayers>` bestaat uit een of meer tags <layer>. Elke tag `<layer>` heeft een attribuut:

- id (elke layer heeft hierin een unieke integer waarde).
- active die de waarde false of true kan hebben: de wfslayer staat standaard aan of uit.
- index die een integerwaarde heeft
- isbasislaag die altijd de waarde false moet hebben
- layer met de externe naam van de wms-laag
- opacity met een integerwaarde tussen 0 en 100
- source met de url waar de laag is op te halen

Achter de tag `<layer>` staat een string (dus geen sql-stament) met de naam zoals de laag op de kaart moet heten: de laagnaam. Deze laagnamen zijn aan en uit te vinken op de kaart in de widget met de titel *filter & kaartlagen* onder het subkopje WMS-lagen. Indien een laagnaam is aangevinkt dan zal Openwave de betreffende externe laag ophalen en tonen.

LET OP:

De WMS-lagen die in de tabel tbgeowms (*beheerportaal tegel: Geo kaartlagen*) aangemerkt zijn

- als WMS-laag: de kolom dlwMS *WMS-laag? (anders WFS)* staat aangevinkt
- EN waarvoor geldt dat de kolom dlinFlexmap (*Altijd opnemen in Flexmap*) ook aangevinkt is

die lagen worden onder water automatisch opgenomen in de xml ook als het blok <wmslayers> niet bestaat.

De objecten van een WMS-laag kunnen in OpenWave geselecteerd worden en die selectie(s) kunnen vervolgens gebruikt worden om bijvoorbeeld een vlak te tekenen.

Blok <widgetbuttons>

voorbeeld widgetbuttons

```
<widgetbuttons>
        <!--er kan een blok opdrachtlinestring, of opdrachtpolygon of
opdrachtpoint volgen -->
        <opdrachtlinestring>
            <button id="1" hint="Lijn opslaan" icoon="177"
action="3">true</button>
            <button id="2" hint="Refresh" icoon="192"
action="4">true</button>
        </opdrachtlinestring>
        <opdrachtpolygon>
            <button id="1" hint="Vlak opslaan" icoon="177"
action="3">true</button>
            <button id="2" hint="Refresh" icoon="192"
action="4">true</button>
        </opdrachtpolygon>
 </widgetbuttons>
```

Het blok <widgetbuttons> mag één keer vookomen (maar hoeft niet). Indien het blok niet voorkomt dan zijn er geen knoppen waarmee een tekenopdracht afgesloten kan worden, terwijl de vierde parameter van de Flexmap-aanroep is gevuld. Binnen het blok <widgetbuttons> mag één keer een blok <opdrachtpoint> en mag één keer een blok <opdrachtlinestring> en mag één keer een blok <opdrachtpolygon> zijn gedefinieerd. In bovenstaand voorbeeld ontbreekt een blok <opdrachtpoint> omdat in de inrichtingstabel alleen een vlak of lijn kan worden opgeslagen.

Binnen een blok <widgetbuttons><opdrachtpoint> of <widgetbuttons><opdrachtpolygon> of <widgetbuttons><opdrachtlinestring> moet één of meer keer de tag <button> voorkomen. Elke tag <opdrachtpolygon><button> heeft een attribuut *id* met een unieke ophogende integer waarde.

Elke tag <opdrachtpoint><button> heeft een attribuut *id* met een unieke ophogende integer waarde. Elke tag <opdrachtlinestring><button> heeft een attribuut *id* met een unieke ophogende integer waarde.
Elke button heeft een attribuut *hint* voor de betreffende knop Elke button heeft een attribuut *icoon* met een integerwaarde dat verwijst naar een OpenWave icoon zie: Iconenlijst

Elke tag <opdrachtpolygon><button> heeft een attribuut *action* met een unieke integer waarde waarnaar wordt verwezen in de blokken <actions><actionpolygon> (zie verderop).

Elke tag <opdrachtpoint><button> heeft een attribuut *action* met een unieke integer waarde waarnaar wordt verwezen in de blokken <actions><actionpoint> (zie verderop). Elke tag <opdrachtlinestring><button> heeft een attribuut *action* met een unieke integer waarde waarnaar wordt verwezen in de blokken <actions><actionlinestring> (zie verderop).

Indien de vierde parameter van de Flexmap aanroep de waarde Polygon heeft, dan kijk OpenWave alleen naar het blok <opdrachtpolygon>. Indien de vierde parameter van de Flexmap aanroep de waarde Point heeft, dan kijk OpenWave alleen naar het blok <opdrachtpoint>. Indien de vierde parameter van de Flexmap aanroep de waarde Point heeft, dan kijk OpenWave alleen naar het blok <opdrachtlinestring>. Indien de vierde parameter van de Flexmap aanroep een lege waarde, dan wordt het hele blok widgetbuttons genegeerd.

De Flexmap heeft zelf ook standaard een aantal knoppen: zie voor de werking daarvan Flexmap knoppen en widgets. Deze zelfgedefineerde knoppen in het blok <widgetbuttons> zijn nodig om

een getekend object op de juiste plaats in de OpenWave database te laten landen.

Let OP: attribuut action opdrachtpoint

Bij een button om een aangewezen punt op te slaan in twee kolommen (dat is meestal het geval in OpenWave: een x-kolom en een y-kolom zoals bijv het centrale punt van een stal) heeft het attibuut *action* twee verwijzingen achter elkaar gescheiden door een komma;

```
<opdrachtpoint>
      <!-- action 1,2 wil zeggen dat de PHP bij deze knop twee acties moet
uitvoeren action 1 en action 2 -->
      <button id="1" hint="Wijzigingen opslaan" icoon="177"
action="1,2">true</button>
      <button id="2" hint="Refresh" icoon="192" action="3">true</button>
</opdrachtpoint>
```

Blok <actions>

voorbeeld actions

```
<actions>
        <actionpolygon>
            <action id="3">setcolumnvalue(tbmilinrichtingen, %keypointer%,
dvgmlpolygoon, {poslist},V))</action>
            <action id="4">refreshActiveDialog</action>
        </actionpolygon>
        <actionlinestring>
            <action id="3">setcolumnvalue(tbmilinrichtingen, %keypointer%,
dvgmlpolygoon, {poslist},V))</action>
            <action id="4">refreshActiveDialog</action>
        </actionlinestring>
</actions>
```

Het blok `<actions>` mag één keer vookomen (maar hoeft niet). Indien het blok niet voorkomt dan kan een tekenopdracht niet afgesloten kan worden (het aangewezen of getekende object wordt dan niet opgeslagen) , terwijl de vierde parameter van de Flexmap-aanroep is gevuld. Binnen het blok `<actions>` mag één keer een blok `<actionpoint>` en mag één keer een blok `<actionlinestring>` en mag één keer een blok `<actionpolygon>` zijn gedefinieerd. In bovenstaand voorbeeld ontbreekt een blok `<actionpoint>` omdat in de inrichtingstabel alleen een vlak of lijn kan worden opgeslagen. Binnen een blok <action><actionpoint> of `<action><actionpolygon>` of `<action><actionlinestring>` moet één of meer keer de tag `<action>` voorkomen.

Elke tag `<actionpolygon><action>` heeft een attribuut *id* met een integer waarde, die correspondeert met een integerwaarde van het attribuut *action* bij de tag `<widgetbuttons><opdrachtpolygon><button>` Zo ook correspondeert attribuut *id* van `<actionlinestring><action>` met het attribuut *action* van de tag

<widgetbuttons><opdrachtlinestring><button> En correspondeert attribuut *id* van <actionpoint><action> met het attribuut *action* van de tag <widgetbuttons><opdrachtpoint><button>

Let Op:attribuut id actiontpoint

Bij een button om een aangewezen punt op te slaan in twee kolommen (dat is meestal het geval in OpenWave: een x-kolom en een y-kolom zoals bijv het centrale punt van een stal) heeft het attibuut *action* twee verwijzingen achter elkaar gescheiden door een komma bijvoorbeeld:

```
<button id="1" hint="Wijzigingen opslaan" icoon="177"
action="1,2">true</button>
```

Dat leidt tot een action definitie voor het opslaan van de X kolom (id = 1) EN eentje voor de Y-kolom (id =2)“:

```
<actionpoint>
       <action id="1">setcolumnvalue(tbmilstal, %keypointer%,
dvxcrd,{pos(x)},V)</action>
       <!-- {pos(x)}  wordt door PHPvervangen door aangwezen x coordinaat -
->
       <action id="2">setcolumnvalue(tbmilstal, %keypointer%,
dvycrd,{pos(y)},V)</action>
       <action id="3">refreshActiveDialog()</action>
</actionpoint>
```

De waarde van de tag <action> moet de aanroep van een OpenWave-action zijn met de juiste parameters (zie ook: Actions) Voor het opslaan van een aangewezen of getekend object moet *setcolumnvalue* worden aangeroepen met de volgende parameters:

- de eerste parameter is de tabelnaam waarin het gegeven moet worden opgelagen.
- de tweede parameter is een verwijzing naar de primary key van die tabel om de juiste rij te vinden.
- de derde parameter is de naam van de kolom waarin het gegeven moet worden opgelagen
- de vierde parameter is de waarde van het gegeven dat moet worden opgeslagen. Daarbij geldt de volgende bijzonderheid bij aanroep vanuit Flexmap:
    - {pos(x)} wordt on the fly vervangen door de waarde van het x-coordinaat-gedeelte van het aangewezen of geslecteerde punt
    - {pos(y)} wordt on the fly vervangen door de waarde van het y-coordinaat-gedeelte van het aangewezen of geslecteerde punt
    - {poslist} wordt on the fly vervangen door de poslist-waarde van het aangewezen of getekende polygoon of lijn.
- de vijfde parameter is de moduleletter voor de rechtenafweging, indien:
    - leeg, dan moet:
        - OF de ingelogde medewerker beheerder zijn (tbmedewerker.dnbeheerniveau > 98) * OF - indien het gaat om tbperceeladressen - het wijzigrecht op tbperceeladressen aangevinkt zijn (tbrechten.dldpcledt)
        - OF de zesde parameter gevuld zijn
    - *V* dan moet de medewerker wijzigrechten hebben op de inrichtingen

(*tbmilrechten.dlbmilinredt*)
    - *W* dan moet de medewerker wijzigrechten hebben op de omgevingzaken (*tbomgrechten.dlbomgedt*)
    - *O* dan moet de medewerker wijzigrechten hebben op de apv.overige zaken (*tbovrechten.dlbovvedt*)
    - *H* dan moet de medewerker wijzigrechten hebben op de handhavingszaken (*tbhhrechten.dlbhahedt*)
- de zesde parameter mag leeg zijn (zoals in alle uitlgeleverde MDMC_*.xml-files) . Indien gevuld gaat deze voor op de vijfde parameter en indien
    - de waarde begint met *tbrechten. of tbomgrechten. of tbovrechten. of tbinforechten. of tbhorrechten. of tbbestrechten. of tbmilvergrechten. of tbmilrechten.* gevolgd door een passende rechtenkolomnaam dan bepaald de evaluatie van die rechtenkolom of de gebruiker voldoende rechten heeft.
    - anders wordt de waarde opgezocht in tbquery (op kolom tbquery.dvcode) en dan bepaald de evaluatie van die query of de gebruiker voldoende rechten heeft

Blok <punten> (OpenWave kaartlagen)

Het blok `<punten>` mag hooguit één keer voorkomen (maar hoeft niet). Binnen het blok `<punten>` bestaan één of meer blokken `<punt>` met een attribuut *id* waarvan de integerwaarde verwijst naar de attribuuten *id* achter de tag `<layer>` in het blok <openwavelayers>. Indien die layer-id de waarde 3 heeft (en aangevinkt is) dan wordt de tags `<punt>` met id-waarde 3 getoond.

Binnen een blok `<punten><punt>` moet één tag `<resultset>` zijn gedefinieerd met als waarde een sqlstatement dat na evaluatie één rij met de kolommen dvcolorname, dvdescription en dvpos bevat (zie hierboven bij `<basisobject type="point">`)

voorbeeld punt

```
<punt layer="3"> <!--rev referentieobjecten punten -->
            <!--kolommen dvcolorname, dvdescription en dvpos zijn verplicht
-->
            <resultset>select a.dvrefcontournaaminrev ||': ' ||
coalesce(a.dvnaamopslag,'') dvdescription, a.dvxcrd
                ||','|| a.dvycrd dvpos, coalesce(a.dvcsscolorname,'Blue')
dvcolorname from vwfrmmilopslag a inner join
                tbmilinrichtingen b on (a.dnkeymilinrichtingen = b.dnkey)
                where
                a.dnkeymilinrichtingen = %keypointer%
                and a.dnkeymilbklactiviteiten is not null
                and a.dvycrd is not null
                and a.dvxcrd is not null
                and a.dvgmlpolygoon is null
            </resultset>
</punt>
```

De evaluatie van de tag `<resultset>` mag null zijn: in dat geval wordt het betreffende point NIET geprojecteerd.

Blok <lijnen> (OpenWave kaartlagen)

Het blok <lijnen> mag hooguit één keer voorkomen (maar hoeft niet). Binnen het blok <lijnen> bestaan één of meer blokken <lijn> met een attribuut *id* waarvan de integerwaarde verwijst naar de attribuuten *id* achter de tag <layer> in het blok <openwavelayers>. Indien die layer-id de waarde 3 heeft (en aangevinkt is) dan wordt de tags <lijn> met id-waarde 3 getoond.

Binnen een blok <lijnen><lijn> moet één tag <resultset> zijn gedefinieerd met als waarde een sqlstatement dat na evaluatie één rij met de kolommen dvstrokecolorname, dvdescription en dvposlist en dfopacity bevat (zie hierboven bij <basisobject type="linestring">)

voorbeeld lijn

```
<lijn layer="3"> <!--rev referentieobjecten lijnen -->
            <!--kolommen dvstrokecolorname, dvdescription en dvposlist zijn
verplicht -->
            <resultset>select a.dvrefcontournaaminrev ||': ' ||
coalesce(a.dvnaamopslag,'') dvdescription, a.dvgmlpolygoon dvposlist,
                coalesce(a.dvcsscolorname,'Blue') dvstrokecolorname from
vwfrmmilopslag a  inner join
                tbmilinrichtingen b on (a.dnkeymilinrichtingen = b.dnkey)
                where a.dnkeymilinrichtingen = %keypointer%
                and a.dnkeymilbklactiviteiten is not null
                and fn_islinestring(a.dvgmlpolygoon) = 1
            </resultset>
</lijn>
```

De evaluatie van de tag <resultset> mag null zijn: in dat geval wordt de betreffende linestring NIET geprojecteerd.

Blok <vlakken> (OpenWave kaartlagen)

Het blok <vlakken> mag hooguit één keer voorkomen (maar hoeft niet). Binnen het blok <vlakken> bestaan één of meer blokken <vlak> met een attribuut *id* waarvan de integerwaarde verwijst naar de attribuuten *id* achter de tag <layer> in het blok <openwavelayers>. Indien die layer-id de waarde 3 heeft (en aangevinkt is) dan wordt de tags <vlak> met id-waarde 3 getoond.

Binnen een blok <vlakken><vlak> moet één tag <resultset> zijn gedefinieerd met als waarde een sqlstatement dat na evaluatie één rij met de kolommen dvfillcolorname, dvstrokecolorname, dvdescription en dvposlist en dfopacity (zie hierboven bij <basisobject type = "polygon">)

Let Op: Radius

De kolom dfradius is optioneel. Indien gevuld dan beschouwt OpenWave deze waarde in meters en wordt het te tekenen polygoon daarmee vergroot

voorbeeld vlak

```
<vlak layer="3"> <!--rev referentieobjecten vlakken -->
            <!--kolommen dvfillcolorname, dvstrokecolorname. dvdescription
en dvposlist en dfopacity en dfradius zijn verplicht -->
            <resultset>select a.dvrefcontournaaminrev ||': ' ||
coalesce(a.dvnaamopslag,'') dvdescription, a.dvgmlpolygoon
                dvposlist, coalesce(a.dvcsscolorname,'Blue')
dvfillcolorname, 'Black' dvstrokecolorname, 0.5 dfopacity, 1 dfradius
                from vwfrmmilopslag a inner join
                tbmilinrichtingen b on (a.dnkeymilinrichtingen = b.dnkey)
                where
                a.dnkeymilinrichtingen = %keypointer%
                and a.dnkeymilbklactiviteiten is not null
                and a.dvgmlpolygoon is not null and
fn_islinestring(a.dvgmlpolygoon) != 1
            </resultset>
</vlak>
```

De evaluatie van de tag `<resultset>` mag null zijn: in dat geval wordt de betreffende polygon NIET geprojecteerd.

Blok <cirkels> (OpenWave kaartlagen)

Het blok `<cirkels>` mag hooguit één keer voorkomen (maar hoeft niet). Binnen het blok `<cirkels>` bestaan één of meer blokken `<cirkel>` met een attribuut *id* waarvan de integerwaarde verwijst naar de attribuuten *id* achter de tag `<layer>` in het blok `<openwavelayers>`. Indien die layer-id de waarde 4 heeft (en aangevinkt is) dan wordt de tags `<cirkel>` met id-waarde 4 getoond.

Binnen een blok `<cirkels><cirkel>` moet één tag `<resultset>` zijn gedefinieerd met als waarde een sqlstatement dat na evaluatie één rij met de kolommen dvfillcolorname, dvstrokecolorname, dvdescription, dvpos, dfradius en dfopacity retourbeert (zie hierboven bij <basisobjecten>). De dfradius is in meters. Er wordt een cirkel van dfradius meters om het punt dvpos geprojecteerd.

voorbeeld cirkel

```
<cirkel layer="4"> <!--REV EVContouren cirkels -->
            <!--kolommen dvfillcolorname, dvstrokecolorname. dvdescription
en dvpos en dfradius en dfopacity zijn verplicht -->
            <resultset>select a.dvevcontournaaminrev || ' '|| coalesce(
a.dv_waarde_3_string,'')
                dvdescription, b.dvxcrd ||','|| b.dvycrd dvpos,
a.df_waarde_2_float dfradius,
                'Yellow' dvfillcolorname, 'Black' dvstrokecolorname, 0.5
dfopacity
                from vwfrmmilopslagevcontour a
                inner join tbmilopslag b
```

```
                on a.dnkeymilopslag = b.dnkey
                inner join tbmilinrichtingen c
                on b.dnkeymilinrichtingen = c.dnkey
                where b.dnkeymilinrichtingen = %keypointer%
                and a.df_waarde_2_float is not null
                and b.dvgmlpolygoon is null
                and b.dvycrd is not null
                and b.dvxcrd is not null
            </resultset>
</cirkel>
```

De evaluatie van de tag `<resultset>` mag null zijn: in dat geval wordt de betreffende cirkel NIET geprojecteerd.

Controle geëvalueerde MDMC_*.xml

Als de instelling *Sectie: Flexmap en Item: opslaanXML* is aangevinkt en er zijn geen fouten opgetreden dan wordt de geëvalueerde versie van de MDMC_*.xml (dus waarin alle SQL-statements zijn geëvalueerd) opgeslagen op de tempmap *TussenMapUploadFiles* die te benaderen is in het beheerportaal: servicecentrum, tegel: Up- en downloadmappen. Bijvoorbeeld onder de naam *5abd522941f1484ba9669e669c77d6c4_MDMC_tbperceeladressen.xml* Dit is uiteindelijk de xml die OpenWave gebruikt om alle geo-informatie te tonen. Daarin is de oorspronkelijke xml uitgebreid met de namespace *xmlns:gml="http://www.opengis.net/gml*".

From:
https://acc-doc.open-wave.nl/ - **Documentatie**

Permanent link:
https://acc-doc.open-wave.nl/oldstable/applicatiebeheer/probleemoplossing/module_overstijgende_schermen/kaart/flexmap_beschrijving?rev=1784106470

Last update: **2026/07/15 11:07**