

# XML-INTERFACE

## View, Muteren en Tariferen

*Een Inleiding tot de module ANVA XML-Interface*

## Inhoudsopgave

<b>Bepalingen</b> .....	<b>4</b>
<b>1 Inleiding</b> .....	<b>5</b>
<b>2 Overzicht functionele entiteitgroepen</b> .....	<b>6</b>
2.1 XML-interface View.....	6
2.2 XML-interface Muteren .....	6
2.3 XML-interface Tariferen .....	7
2.4 XML-Interface View en Muteren Entiteiten .....	8
<b>3 Overzicht functionele entiteitgroepen</b> .....	<b>9</b>
3.1 XML-interface View.....	9
3.2 XML-interface Muteren .....	9
3.3 XML-interface Tariferen .....	10
<b>4 ANVA AWACS</b> .....	<b>11</b>
4.1 Character set Encoding .....	11
4.2 <AWACS> .....	12
4.2.1 <HEADER>.....	12
4.2.2 <SERVICE_ID> .....	12
4.2.3 <DATASOURCE>.....	12
4.2.4 <KEY>.....	13
4.2.5 <DATA> .....	13
<b>5 XML-Interface-View</b> .....	<b>14</b>
5.1 <REQUEST> .....	14
5.1.1 <FUNCTION> .....	14
5.1.1.1 <PARAM>.....	15
<b>6 XML-Interface-View Entiteiten</b> .....	<b>17</b>
6.1 Collecties .....	17
6.2 Entiteiten.....	17
6.2.1 Sleutel <PK>.....	17
6.2.2 Label <L> .....	17
6.2.3 Index <IND>.....	18
6.3 Selectie <WHERE> .....	18
6.3.1 Sleutel <PK>, Label <L> , Index <IND> en operators .....	18
6.3.2 Vergelijkingen en operators .....	19
6.3.3 Beperking TOP .....	20
6.3.4 Startpunt in combinatie met TOP.....	20
6.4 Hiërarchie entiteiten.....	21
6.5 Response .....	22
<b>7 XML-Interface-Muteren</b> .....	<b>23</b>
7.1 WebMuteren.update .....	23
7.2 XML-Interface Muteren DTD.....	23
7.3 Parameters .....	24
<b>8 XML-Interface-Muteren Entiteiten</b> .....	<b>25</b>
8.1 Collecties .....	25
8.2 Update .....	26
8.2.1 ADN View.....	26
8.3 Gebruik van ACTIE .....	26
8.4 Nieuwe Records .....	28
8.5 Execute.....	28
8.6 Result .....	28
8.6.1 DTD Result .....	28
<b>9 XML-Interface voorbeelden</b> .....	<b>30</b>
9.1 Mutaties in de Wacht .....	30

9.1.1	Wachtmutatie op bestaande polis.....	30
9.1.2	Mutatie van bestaande dekking.....	30
9.2	Verwijderen Dekking.....	31
9.2.1	Verwijderen Dekking.....	31
9.2.2	Dekking verwijderen in de Wacht .....	32
9.3	Mandaatgegevens (SEPA) .....	33
9.3.1	Mandaatgegevens opvragen .....	33
9.3.2	Mandaatgegevens opvragen zonder kantoorcode (geen concern of concern met één boekhouding) .....	33
9.3.3	Mandaatgegevens muteren .....	34
9.3.4	Mandaatgegevens muteren zonder kantoorcode (geen concern of concern met één boekhouding) .....	34
9.4	Nieuwe relatie opvoeren .....	35
9.4.1	Object opvoeren bij bestaande relatie .....	35
9.4.2	Bestaand object bij een relatie muteren .....	35
9.4.3	Compliance-check uitvoeren.....	36
9.5	Verwijzing opgeven.....	37
9.6	Formulier genereren bij muteren of invoeren van een (wacht)polis .....	38

© 2022, ANVA bv  
Uitgegeven in eigenbeheer  
[www.anva.nl](http://www.anva.nl)

Alle rechten voorbehouden.

Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand en/of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, door fotokopie, opnamen of op enige ander manier zonder voorafgaand schriftelijke toestemming door ANVA bv.

## Bepalingen

### Gebruik

De XML-documentatie mag enkel en alleen gebruikt worden voor XML-entiteiten waar licenties voor zijn aangeschaft bij ANVA bv. De documentatie mag enkel en alleen door de licentiehouder van de XML-entiteiten beschikbaar worden gesteld aan een derde partij, mits deze derde partij uit naam van de licentiehouder XML-georiënteerde oplossingen bouwt. ANVA houdt de licentiehouder verantwoordelijk dat de XML-documentatie door de derde partij enkel en alleen voor de licentiehouder wordt aangewend en er geen reclame-uitingen mee worden verricht door de derde partij. ANVA behoudt zich het recht voor om bij overschrijding van deze gebruiksvoorwaarden de XML-licentie per direct in te trekken.

### Inzet ANVA XML-Interface

Met betrekking tot de ANVA XML-Interface is van belang te melden dat deze interface is afgestemd op de ANVA-programmatuur en ook louter functioneert in combinatie met deze software op het gebied van relatie-, polis- en dekkingsgegevens. Met behulp van de XML-Interface kunnen uw klanten en of Agenten via uw webapplicatie relatie- en polisgegevens uit de ANVA raadplegen en muteren.

De ANVA XML-Interface is nadrukkelijk bedoeld om de ANVA-gegevens te ontsluiten voor uw klanten. Indien de XML-Interface wordt aangewend voor andere doeleinden, zoals inzetten van andere applicaties voor de ANVA-gebruikers met mede tot doel om de ANVA-gebruikerslicentie terug te brengen, anderzijds indien het tot doel heeft de ANVA-applicatie te ontsluiten voor aangesloten Agenten en of anderen type gebruikers, dit om op die manier de ANVA-applicatie te hosten als zijnde een provider, dan geldt er voor het gebruik van de ANVA-applicatie een ander licentiemodel. U dient vooraf aan ANVA kenbaar te maken met welk doel u de XML-Interface gaat inzetten. Een aangepast ANVA-applicatie licentiemodel kan dan voor uw kantoor van toepassing zijn.

De ANVA XML-Interface, in de huidige versie, werkt alleen met gegevens die worden gebruikt en beheert binnen de isCOBOL-runtime en Ctree Fileserver.

### Oneigenlijk gebruik

Indien ANVA constateert dat er oneigenlijk gebruik en of misbruik wordt/is gemaakt van de door haar aan licentiehouder verstrekte XML-Interface, de XML-entiteiten en of XML-documentatie, dan behoudt ANVA zich het recht voor om de XML-licentie per direct in te trekken en als nog de licentiekosten aan de licentiehouder door te belasten.

### Disclaimer

Ondanks dat de XML-Interface, haar XML-entiteiten en bijbehorende XML-documentatie door ANVA uitvoerig zijn getest en gecontroleerd, kan ANVA niet uitsluiten dat er zich onvolkomenheden kunnen voordoen bij gebruik van de algehele XML-Interface. ANVA kan nimmer aansprakelijk gesteld worden, in enige vorm dan ook, voor mogelijke bedrijfsschade en/of verlies die de licentiehouder mogelijkerwijs ondervindt of ondervonden heeft door het gebruik van de XML-Interface.

## 1 Inleiding

Met de modules XML-Interface-View, XML-Interface-Muteren en XML-Interface-Tarifieren kunnen gegevens door middel van XML requests uit een ANVA dataset worden opgevraagd, dit gaat met behulp van een HTTP Post request.

Om gebruik van de XML-Interface te kunnen maken moet de ANVA Business Server (ABS) met ANVA AWACS geïnstalleerd zijn.

## 2 Overzicht functionele entiteitgroepen

Binnen de **XML-Interface onderscheiden we functionele entiteitgroepen. De entiteitgroepen zijn logisch opgebouwd uit gerelateerde functies. Entiteitgroepen overlappen deels en vullen elkaar aan. Afhankelijk van de toepassing kan er een keus uit de volgende functionele entiteitgroepen gemaakt worden, welke separaat aangeschaft dienen te worden.**

### 2.1 XML-interface View

XML-Interface-View	Entiteiten	Extra index mogelijk
Relatie-view	Relatie	ja
Contracten-view	Relatie-info	
	Pakketten	ja
	Polissen	ja
	Dekkingen	ja
	Pool-verdeling	nee
	Assuradeuren-verdeling	nee
Dossier-view	Wacht polis	ja
	Wacht dekking	ja
Dossier-view	Dossier	nee
	PDF's	nee
Agenda/WKL-view	Agenda	nee
	Werklijst	nee
Contact-view	Contact-administratie	nee
Factuur-view	Factuur	ja
	Facspect	nee
	Betalingen	ja
Schade-view	Schade	ja
	Schadepolis	ja
	Schadedekking	ja
	Schadebetrokkene	nee
	Online gegenereerd formulier in XML.	nee

Een toelichting op het gebruik van "Extra index" is opgenomen in § 6.2.3. Index IND

### 2.2 XML-interface Muteren

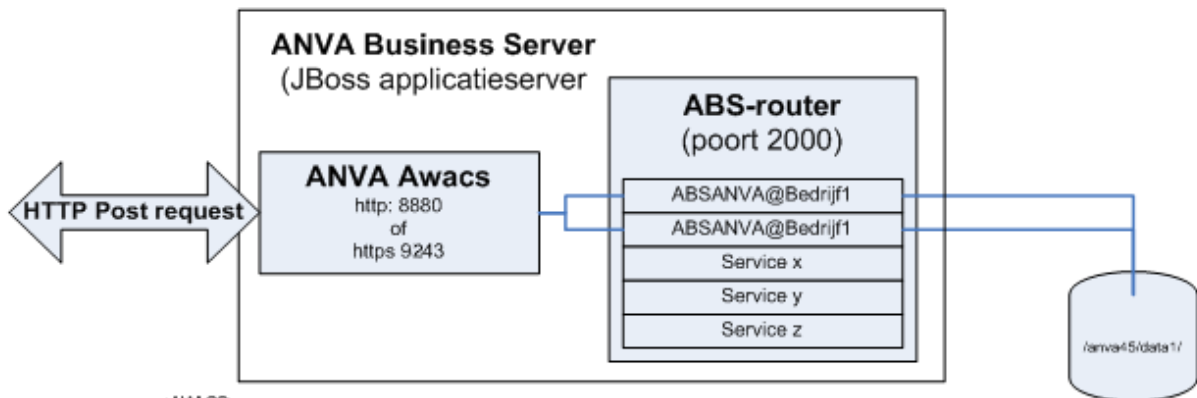
XML-Interface-Muteren	Entiteiten
Relatie-mut	Relatie
Contracten-mut	Pakketten
	Polissen
	Dekkingen
	Wacht polis
	Wacht dekking
Agenda/WKL-mut	Agenda
	Werklijst
Contact-mut	Contact-administratie
Schade-mut	Schade gegevens algemeen
	Schade relatie
	Schadepolis
	Schade dekking
	Vluchtige betrokkene
	Koppelen vaste betrokkene
	Schadememo
	Schadeoorzaak
	Schadereserve
	Koppeling extern document
Schade-boeking-mut	Schadeboeking
	Schade uitkering in de wacht
Dienstverband-mut	Dienstverband

### 2.3 XML-interface Tariferen

XML-Interface-Tariferen	Entiteiten
Tariferen	Pakket Polis Dekking

ANVA Aln dit document wordt de XML communicatie van XML-Interface beschreven. We kunnen dit splitsen in elementen voor de ANVA Awacs, XML-Interface-View en XML-Interface-View-Entiteiten. De specifieke aspecten voor de XML-Interface-Muteren worden in een apart document beschreven.

Schematische weergave van de XML-Interface:



```

<AWACS>
  <HEADER>
    <SERVICE_ID>ABSANVA</SERVICE_ID>
    <DATASOURCE>Bedrijf1</DATASOURCE>
    <KEY>... </KEY>
  </HEADER>
  <DATA>
    <REQUEST>
      <FUNCTION NAME="webview.select">
        <PARAM NAME="select">
          <RELATES>
            <RELATE>
              <PK/>
              <LN NR="*" />
            </RELATE>
            <WHERE>
              <LN NR="2" EQ="Test naamf" />
            </WHERE>
          </RELATES>
        </PARAM>
      </FUNCTION>
    </REQUEST>
  </DATA>
</AWACS>

<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATES>
        <RELATE>
          <PK/>
          <LN NR="*" />
        </RELATE>
        <WHERE>
          <LN NR="2" EQ="Test naamf" />
        </WHERE>
      </RELATES>
    </PARAM>
  </FUNCTION>
</REQUEST>

```

## 2.4 XML-Interface View en Muteren Entiteiten

De XML-Interface is opgebouwd uit 2 hoofdgroepen, XML-Interface-View en XML-Interface-Muteren. Binnen deze 2 hoofdgroepen zijn weer functionele entiteitgroepen te onderscheiden. Om met de XML-Interface te kunnen werken, dient u minimaal in bezit te zijn van de XML-Interface-View standaard functionele entiteitgroep. In dit document worden voorbeelden gebruikt gebaseerd op de XML-Interface-View entiteitgroepen. De opbouw van de XML-definities voor de XML-Interface-Muteren zijn identiek.



### 3 Overzicht functionele entiteitgroepen

Binnen de XML-Interface onderscheiden we functionele entiteitgroepen. De entiteitgroepen zijn logisch opgebouwd uit gerelateerde functies. Entiteitgroepen overlappen deels en vullen elkaar aan. Afhankelijk van de toepassing kan er een keus uit de volgende functionele entiteitgroepen gemaakt worden, welke separaat aangeschaft dienen te worden.

#### 3.1 XML-interface View

XML-Interface-View	Entiteiten	Extra index mogelijk
Relatie-view	Relatie	ja
Contracten-view	Relatie-info	
	Pakketten	ja
	Polissen	ja
	Dekkingen	ja
	Pool-verdeling	nee
	Assuradeuren-verdeling	nee
Dossier-view	Wacht polis	ja
	Wacht dekking	ja
Dossier-view	Dossier	nee
	PDF's	nee
Agenda/WKL-view	Agenda	nee
	Werklijst	nee
Contact-view	Contact-administratie	nee
Factuur-view	Factuur	ja
	Facspect	nee
	Betalingen	ja
Schade-view	Schade	ja
	Schadepolis	ja
	Schadedekking	ja
	Schadebetrokkene	nee
	Online gegenereerd formulier in XML.	nee

Een toelichting op het gebruik van "Extra index" is opgenomen in § 6.2.3. Index IND

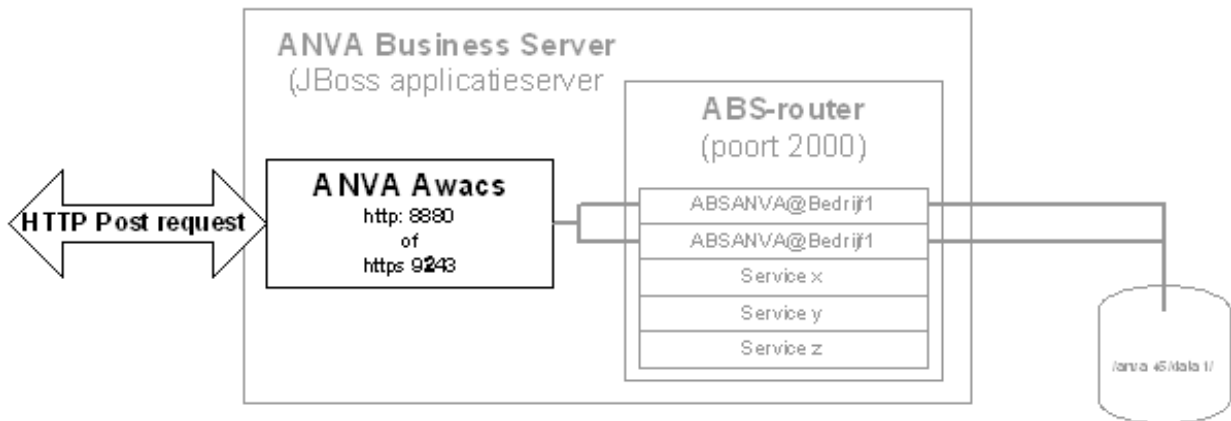
#### 3.2 XML-interface Muteren

XML-Interface-Muteren	Entiteiten
Relatie-mut	Relatie
Contracten-mut	Pakketten
	Polissen
	Dekkingen
	Wacht polis
	Wacht dekking
Agenda/WKL-mut	Agenda
	Werklijst
Contact-mut	Contact-administratie
Schade-mut	Schade gegevens algemeen
	Schade relatie
	Schadepolis
	Schade dekking
	Vluchtige betrokkene
	Koppelen vaste betrokkene
	Schadememo
	Schadeoorzaak
	Schadereserve
	Koppeling extern document
Schade-boeking-mut	Schadeboeking
	Schade uitkering in de wacht
Dienstverband-mut	Dienstverband

### 3.3 XML-interface Tariferen

XML-Interface-Tariferen	Entiteiten
Tariferen	Pakket Polis Dekking

## 4 ANVA AWACS



ANVA AWACS is een HTTP Post request service. ANVA AWACS stuurt het uiteindelijke verzoek door via ABS-router aan de juiste ABSANVA. ABSANVA leest uiteindelijk de gegevens uit de ANVA4/5 database.

Het HTTP Post request moet worden gestuurd naar <http://IPAdres:8180/anva/awacs/absagent>

Het IP adres en poort zijn van de ANVA Business Server waar ANVA AWACS beschikbaar is. In een HTML formulier zou dat als volgt eruit zien:

```
<form action="http://localhost:8180/anva/awacs/absagent" method="post">
<textarea name="request">
  <AWACS>
    <HEADER>
      <SERVICE_ID>ABSANVA</SERVICE_ID>
      <DATASOURCE>Bedrijf1</DATASOURCE>
      <KEY>...</KEY>
    </HEADER>
    <DATA>
      <REQUEST>
        ...
      </REQUEST>
    </DATA>
  </AWACS>
</textarea>
<input type="submit">
</form>
```

### 4.1 Character set Encoding

De XML declaratie is optioneel en moet aan het begin van de XML staan (voor <AWACS>). De character set encoding moet dan hetzelfde zijn als de encoding die in de HTTP headers is opgegeven.

Indien de character set niet aan elkaar gelijk zijn, dan is met name de uitkomst voor diakritische tekens ongedefinieerd. Een é kan dan bijvoorbeeld worden weergegeven als een ◻. ANVA AWACS vertaalt zelf de encoding naar de IBM858 character set.

## 4.2 <AWACS>

Het root element voor een verzoek aan ANVA AWACS is: <AWACS>.

```
<AWACS>
  <HEADER>
    <SERVICE_ID>ABSANVA</SERVICE_ID>
    <DATASOURCE>Bedrijf1</DATASOURCE>
    <KEY>...</KEY>
  </HEADER>
  <DATA>
    <REQUEST>
      ...
    </REQUEST>
  </DATA>
</AWACS>
```

### 4.2.1 <HEADER>

In het <HEADER> element moeten altijd de elementen <SERVICE\_ID> en <DATASOURCE> voorkomen. Het element <KEY> is optioneel.

```
<AWACS>
  <HEADER>
    <SERVICE_ID>ABSANVA</SERVICE_ID>
    <DATASOURCE>Bedrijf1</DATASOURCE>
    <KEY>...</KEY>
  </HEADER>
  <DATA>
    <REQUEST>
      ...
    </REQUEST>
  </DATA>
</AWACS>
```

### 4.2.2 <SERVICE\_ID>

Het element <SERVICE\_ID> heeft altijd de waarde ABSANVA.

```
<AWACS>
  <HEADER>
    <SERVICE_ID>ABSANVA</SERVICE_ID>
    <DATASOURCE>Bedrijf1</DATASOURCE>
    <KEY>...</KEY>
  </HEADER>
  <DATA>
    <REQUEST>
      ...
    </REQUEST>
  </DATA>
</AWACS>
```

### 4.2.3 <DATASOURCE>

Het element <DATASOURCE> zal als ANVA4/5 op een Windows Server staat de waarde "Bedrijf1" en bij ANVA4/5 op een Unix server de waarde "1" hebben.

Bij gebruik van meerdere ANVA4/5 datasets kan dit wijzigen in "Bedrijf2" cq "2".

```
<AWACS>
  <HEADER>
    <SERVICE_ID>ABSANVA</SERVICE_ID>
    <DATASOURCE>Bedrijf1</DATASOURCE>
    <KEY>...</KEY>
  </HEADER>
  <DATA>
```

```
<REQUEST>
...
</REQUEST>
</DATA>
</AWACS>
```

#### 4.2.4 <KEY>

Als in de ANVA AWACS configuratie (“/anva-abs/jboss-3.2.5/server/anva-01/conf/anva/awacs.xml”) een key is ingevuld, zal de key ook in het <KEY> element moeten worden ingevuld. Hierdoor kan alleen met een geldige key ANVA AWACS worden aangeroepen.

```
<AWACS>
  <HEADER>
    <SERVICE_ID>ABSANVA</SERVICE_ID>
    <DATASOURCE>Bedrijf1</DATASOURCE>
    <KEY>...</KEY>
  </HEADER>
  <DATA>
    <REQUEST>
      ...
    </REQUEST>
  </DATA>
</AWACS>
```

Als de key niet is ingevuld of niet overeenkomt is de response HTTP Status 403.

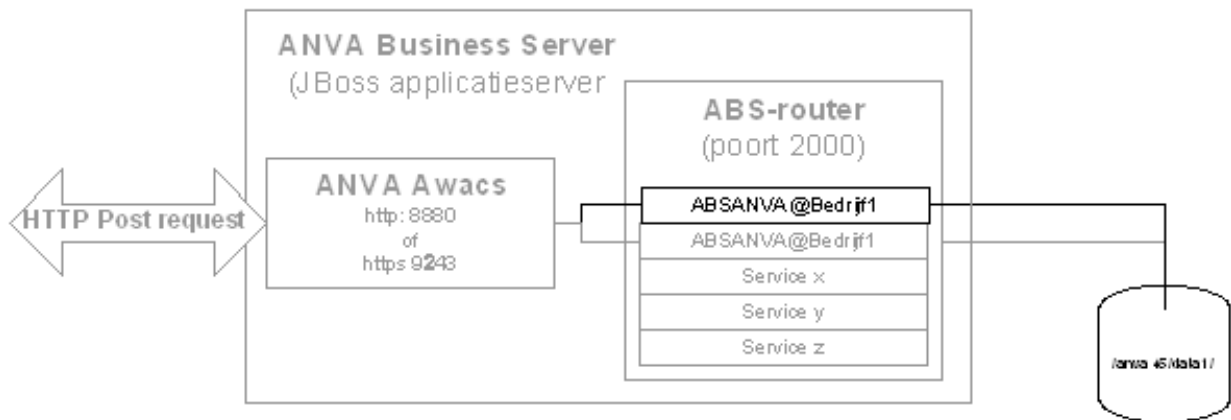
#### 4.2.5 <DATA>

In het element <DATA> staat het verzoek wat aan ABSANVA wordt doorgestuurd. In het hoofdstuk XML-Interface-View en in het hoofdstuk XML-Interface-View Entiteiten wordt het request wat in het <DATA> element moet komen verder beschreven.

```
<AWACS>
  <HEADER>
    <SERVICE_ID>ABSANVA</SERVICE_ID>
    <DATASOURCE>Bedrijf1</DATASOURCE>
  </HEADER>
  <DATA>
    <REQUEST>
      ...
    </REQUEST>
  </DATA>
</AWACS>
```

## 5 XML-Interface-View

De XML-Interface-View request wordt opgebouwd uit een Function gevolgd door een Parameter.



XSD (XML Schema Definition Language) zijn niet beschikbaar.

### 5.1 <REQUEST>

Het root element voor een verzoek aan ABSANVA is: <REQUEST>.

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="*" />
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

#### 5.1.1 <FUNCTION>

Voor XML-Interface-View heeft het element <FUNCTION> een attribuut NAME met de waarde "webview.select".

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="*" />
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

### 5.1.1.1 <PARAM>

Voor XML-Interface-View heeft het element <PARAM NAME> een attribuut "select".

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="*" />
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

Binnen het element <FUNCTION NAME="webview.select"> zijn de volgende parameters optioneel.

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="stats"/>
    <PARAM NAME="display"/>
    <PARAM NAME="beknopt"/>
    <PARAM NAME="oms"/>
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="*" />
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

- **<PARAM NAME="stats"/>**

Optioneel. Indien <PARAM NAME="stats"/> aanwezig is worden er enkele statistieken zoals <RESPONSE\_TIME> van de query teruggegeven.

- **<PARAM NAME="display"/>**

Optioneel. Indien <PARAM NAME="display"/> aanwezig is zal naast de waarde de geformatteerde waarde (displaywaarde), zoals in ANVA4/5 pad BLL (Beheer | Labels en coderingen | Labels) staat, als attribuut in de teruggegeven labels worden opgenomen.

- **<PARAM NAME="beknopt"/>**

Optioneel. Indien <PARAM NAME="beknopt"/> aanwezig is zal de **Omschrijving beknopt**, zoals in ANVA4/5 pad BLL (Beheer | Labels en coderingen | Labels) staat, als attribuut in de teruggegeven labels worden opgenomen.

- **<PARAM NAME="oms"/>**

Optioneel. Indien <PARAM NAME="oms"/> aanwezig is zal de **Omschrijving**, zoals in ANVA4/5 pad BLL (Beheer | Labels en coderingen | Labels) staat, als attribuut in de teruggegeven labels worden opgenomen.

Hieronder een voorbeeld response van de attributen "display", "beknopt" en "oms" en een voorbeeldscherm in ANVA4/5 pad BLL.

```
<L NR="29" OMS="Geboortedatum" BEKNOPT="Geb.dat." DISPLAY="12-10-1970">19701012</L>
```

is Label ✕

Help

? 🏠 ✕ !

Label	<input type="text" value="29"/>
Omschrijving	<input type="text" value="Geboortedatum"/>
Omschrijving beknopt	<input type="text" value="Geb.dat."/>
Code	<input type="radio"/> Ja <input checked="" type="radio"/> Nee
Accept/display wijze	<input type="text" value="Datum (DD-MM-EEJJ, &lt;= sys-datum)"/>
Lengte	<input type="text" value="10"/>
Betekenis	<input type="text"/>

Valuta/euro afh. ?  Ja  Nee



## 6 XML-Interface-View Entiteiten

Binnen het element `<PARAM NAME="select">` wordt de XML-Interface-View opdracht opgegeven. Deze entiteiten kunnen los of in een hiërarchie worden opgevraagd.

### 6.1 Collecties

Met een collectie element (`<RELATIES>`, `<PAKKETTEN>`, `<POLISSEN>`, enz.) wordt aangegeven uit welke tabel er iets teruggegeven moet worden.

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="*" />
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

### 6.2 Entiteiten

Met een entiteit element (`<RELATIE>`, `<PAKKET>`, `<POLIS>`, enz.) wordt aangegeven wat er uit elke rij van die tabel teruggegeven moet worden.

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="*" />
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

In bovenstaand voorbeeld wordt de ANVA4/5 entiteit `<RELATIE>` gebruikt. Dezelfde syntax wordt gebruikt voor de overige entiteiten. De entiteit is verplicht, de inhoud is optioneel.

#### 6.2.1 Sleutel `<PK>`

Met `<PK/>` wordt gespecificeerd dat de primary key van het record moet worden teruggegeven. Dit veld is nodig wanneer deze specifieke relatie opnieuw opgehaald moet worden.

#### 6.2.2 Label `<L>`

Met `<L NR="..." />` kan een label worden gespecificeerd dat moet worden teruggegeven. Het label wordt ook teruggegeven wanneer het een lege waarde heeft. U kunt meerdere labels `<L NR="..." />` onder elkaar worden opgegeven.

Met `<L NR="*" />` kan worden gespecificeerd dat alle gevulde labels moeten worden teruggegeven. Een veld wordt, ook wanneer het met meerdere labels op te halen is slechts 1 keer teruggegeven.

### 6.2.3 Index <IND>

Met de optie <IND NAME="..." /> kan in bepaalde bestanden een indexveld worden gespecificeerd dat moet worden teruggegeven. Het veld wordt ook teruggegeven wanneer het een lege waarde heeft.

Met <IND NAME="\*" /> kan worden gespecificeerd dat alle gevulde indexvelden moeten worden teruggegeven.

Het zoekbestand met de extra index(en) kan met behulp van een ANVAINI-bestand en via pad IIRDO worden opgebouwd.

## 6.3 Selectie <WHERE>

Met de optie WHERE element kan aangegeven worden welke records er geselecteerd moeten worden.

Het WHERE element kan meerdere criteria bevatten. De criteria zijn eenvoudig van opzet en moeten allemaal TRUE opleveren voor de selectie van een record (boolse en operator).

Indien het eerste label van de WHERE-clausule een bestaande index van het bestand betreft (bijvoorbeeld agent in het polisbestand) zal XML-Interface-View deze index gebruiken. Hierdoor kan het resultaat aanzienlijk sneller aangeleverd worden. Het resultaat zal op de volgorde van deze index binnenkomen.

Indien het eerste label van de WHERE-clausule een alternatieve extra index van het bestand betreft zal XML-Interface-View deze index gebruiken. Hierdoor kan het resultaat aanzienlijk sneller aangeleverd worden. Het resultaat zal dan echter op de volgorde van deze index binnenkomen. In een tabel in hoofdstuk 3.1 is aangegeven voor welke labels een extra index mogelijk is.

Indien het eerste label van de WHERE-clausule geen index van het bestand is, dan kan de selectie langer duren.

### 6.3.1 Sleutel <PK>, Label <L> , Index <IND> en operators

Er kan geselecteerd worden op de primary key (<PK EQ="..." />) de labels (<L NR="..." EQ="..." />) en een extra indexveld (<IND NAME="..." EQ="..." />).

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="2"/>
          <POLISSEN>
            <POLIS>
              <PK/>
              <L NR="1"/>
              <L NR="10000"/>
              <L NR="10004"/>
            </POLIS>
          </POLISSEN>
        </RELATIE>
        <WHERE>
          <L NR="10027" EQ="123"/>
        </WHERE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

In bovenstaand voorbeeld worden één relatielabel en drie polislabels opgevraagd van alle polissen die in label 10027 - Agent de waarde 123 hebben.

```
<RELATIES>
  <RELATIE>
```

```

<PK/>
<L NR="2"/>
<POLISSEN>
  <POLIS>
    <PK/>
    <L NR="1"/>
    <L NR="10000"/>
    <L NR="10004"/>
  </POLIS>
</POLISSEN>
</RELATIE>
<WHERE>
  <IND NAME=ZOEK-10017 = EQ="1"/>
</WHERE>
</RELATIES>
</PARAM>
</FUNCTION>
</REQUEST>

```

In dit voorbeeld worden één relatielabel en drie polislabele opgevraagd van alle polissen die in extra indexlabel 10017 – Wijzigingsreden de waarde 1 hebben. Het zoekbestand met de extra index(en) kan met behulp van een ANVAINI-bestand en via pad IIRDO worden opgebouwd.

### 6.3.2 Vergelijkingen en operators

Bij de selectie-criteria op labels onder de WHERE-clausule kunnen de volgende vergelijkingen opgegeven worden:

vergelijking	Operator(s)	XML
Gelijk aan	a = b	<L NR="..." EQ="b">
niet gelijk aan	a ≠ b	<L NR="..." NE="b">
kleiner dan	a < b	<L NR="..." LT="b">
groter dan	a > b	<L NR="..." GT="b">
kleiner dan of gelijk aan	a ≤ b	<L NR="..." LE="b">
groter dan of gelijk aan	a ≥ b	<L NR="..." GE="b">
Bereik	b ≤ a ≤ c	<L NR="..." GE="b" LE="c">

vergelijking	Operator(s)	XML
Gelijk aan	label = waarde	<L NR="..." EQ="waarde">
niet gelijk aan	label ≠ waarde	<L NR="..." NE="waarde">
kleiner dan	label < waarde	<L NR="..." LT="waarde">
groter dan	label > waarde	<L NR="..." GT="waarde">
kleiner dan of gelijk aan	label ≤ waarde	<L NR="..." LE="waarde">
aan		
groter dan of gelijk aan	label ≥ waarde	<L NR="..." GE="waarde">
Bereik	waarde ≤ label ≤ waarde	<L NR="..." GE="waarde" LE="waarde">

Voor een niet inclusief bereik (waarde < label < waarde) kan ook <L NR="..." GT="b" LT="c"> gebruikt worden. Ook (waarde ≤ label < waarde) en (waarde < label ≤ waarde) kunnen op deze wijze aangegeven worden. Wanneer de vergelijking <L NR="..." GT="b" LT="c"> niet werkt, kan deze als volgt genoteerd worden:

```

<L NR="..." GT="b">
<L NR="..." LT="c">

```

Alle vergelijkingen zijn hoofdletterongevoelig (case-insensitive).

XML-Interface-View kan in ieder geval 5 criteria per collectie aan (dit kunnen er meer zijn). Wanneer het maximum aantal criteria overschreden is wordt er een foutmelding teruggegeven.

De vergelijkingen worden in het ANVA systeem door dezelfde routine uitgevoerd als in het selectie systeem. Dit betekent onder andere dat er in de waarde wildcards gebruikt kunnen worden:

Wildcard	Beschrijving
*	0, 1 of meer willekeurige karakters
?	1 willekeurig karakter

### 6.3.3 Beperking TOP

De resultaatset kan beperkt worden in het aantal op te leveren entiteiten met de optie (<WHERE TOP="11"/>). In dit voorbeeld worden er maximaal 11 entiteiten teruggegeven.

```
<RELATIES>
  <POLISSEN>
    <POLIS>
      <PK/>
      <L NR="1"/>
      <L NR="10000"/>
      <L NR="10004"/>
    </POLIS>
  </POLISSEN>
  <WHERE TOP="11">
    <L NR="10027" EQ="00123"/>
  </WHERE>
</RELATIES>
</PARAM>
</FUNCTION>
</REQUEST>
```

### 6.3.4 Startpunt in combinatie met TOP

```
<WHERE TOP="11">
  <L NR="10027" EQ="123"/>
  <PK GT="POLBES_00000456020500105001"/>
</WHERE>
</RELATIES>
</PARAM>
</FUNCTION>
</REQUEST>
```

Indien de vorige bevraging middels een TOP-opdracht een beperkte resultaatset heeft opgeleverd, dan kan met een regel

```
<PK GT="POLBES_000004560500105001"/>
```

een startpunt worden opgegeven op de primary key binnen de reeks van de eerste sleutel (wijzigingsreden = 123). Hierbij wordt direct gestart binnen de geselecteerde key in de eerste regel (wijzigingsreden in dit voorbeeld) op het record met de poliskey 000000020500105001 en niet met de kleinste poliskey. Het startpunt voor de tweede serie is hier het laatste polis-keynummer; die key moet door de aanvrager worden meegegeven in de <PK GT

Agentkey	Poliskey
123	000000050200102001
123	000000050200302003
123	000000050300102001
123	000000080200102001
123	000000080200202002
123	000000080300102001
123	000001290100101001
123	000001290200102001
123	000001290200202002
123	000001290800108001
123	000004560200102001

TOP = 11 ,

123	000004560200102001
123	000004560200402004

```
123          000004880100101001
123          000004880200102001
enz
```

## 6.4 Hiërarchie entiteiten

Entiteiten kunnen in de XML in een hiërarchie geplaatst worden. Hiervoor wordt naast de labels van een entiteit een collectie element geplaatst waar de onderliggende entiteiten in geplaatst kunnen worden:

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="select">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="*" />
          <POLISSEN>
            <POLIS>
              <PK/>
              <L NR="*" />
            </POLIS>
          </POLISSEN>
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

Iedere entiteit kan op deze manier in principe in iedere andere entiteit genest worden.

Wanneer aan XML-Interface-View een hiërarchie van entiteiten wordt opgevraagd zullen sommige combinaties echter geen geneste entiteiten opleveren.

In de entiteit kunnen andere collectie elementen, al dan niet met `WHERE` elementen, worden genest. Hierbij wordt rekening gehouden met de impliciete relatie tussen de geneste entiteit en zijn parent.

```
<RELATIES>
  <RELATIE>
    <PK/>
    <L NR="*" />
    <POLISSEN>
      <POLIS>
        <PK/>
        <L NR="*" />
      </POLIS>
    </POLISSEN>
  </RELATIE>
  <WHERE>
    <L NR="2" EQ="Jansen" />
  </WHERE>
</RELATIES>
```

## 6.5 Response

Het antwoord op de eenvoudige relatie selectie ziet er ongeveer als volgt uit:

```
<?xml version="1.0" encoding="IBM858"?>
<RESPONSE>
  <RESULT NAME="webview.select" CODE="0">
    <RELATIES>
      <RELATIE>
        <PK>...</PK>
        <L NR="1">...</L>
        <L NR="2">...</L>
        <L NR="3">...</L>
        <L NR="4">...</L>
        ...
      </RELATIE>
    </RELATIES>
  </RESULT>
</RESPONSE>
```

De volgorde waarop de labels worden teruggegeven is niet gedefinieerd.

De volgorde waarop de entiteiten in de collecties worden teruggegeven is ook niet gedefinieerd.

De volgorde waarop geneste collecties worden teruggegeven is hetzelfde als waarop deze werden aangevraagd.

Als in de REQUEST om een geneste collectie is gevraagd maar er zijn geen geneste items gevonden dan wordt dit aangegeven met een lege geneste collectie:

```
<?xml version="1.0" encoding="IBM858"?>
<RESPONSE>
  <RESULT NAME="webview.select" CODE="0">
    <RELATIES>
      <RELATIE>
        <PK>...</PK>
        <L NR="...">...</L>
        ...
        <POLISSEN/>
      </RELATIE>
    </RELATIES>
  </RESULT>
</RESPONSE>
```

## 7 XML-Interface-Muteren

De XML-Interface-Muteren kan enkel worden toegepast als men ook in het bezit is van de XML-Interface-View.

### 7.1 WebMuteren.update

In XML-Interface-Muteren wordt voor het doorvoeren van een mutatie de functie: WebMuteren.update gebruikt.

Een eenvoudige relatie mutatie ziet er als volgt uit:

```
<?xml version="1.0" encoding="IBM858"?>
<REQUEST>
  <FUNCTION NAME="webmuteren.update">
    <PARAM NAME="view">
      <RELATIES>
        <RELATIE>
          <PK/>
          <L NR="...">...</L>
          ...
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

**Notitie** Omdat de XML naar een ANVA programma gestuurd wordt dat met de Nederlandse tekenset werkt (OS afhankelijk) wordt de XML altijd met de encoding "IBM858" geïnterpreteerd.

### 7.2 XML-Interface Muteren DTD

De Data Type Definition (DTD) voor dit bericht is als volgt:

```
<?xml version="1.0" encoding="ibm858"?>
<!ELEMENT REQUEST (FUNCTION)>

<!ELEMENT FUNCTION (PARAM)+>
<!ATTLIST FUNCTION (NAME (webmuteren.update) #REQUIRED|TIMEOUT (UTC)|VALIDATE (true|false))>

<!ELEMENT PARAM (RELATIES | PAKKETTEN | POLISSEN | DEKKINGEN
| WACHTPOLISSEN | WACHTDEKKINGEN | SCHADES
| CONTACTEN)+>
<!ATTLIST PARAM NAME (verbose | debug | execute | view | login) #REQUIRED>

<!-- View -->
<!ELEMENT RELATIES (RELATIE)+>
<!ELEMENT RELATIE (PK?, L*, (PAKKETTEN | POLISSEN | WACHTPOLISSEN
| CONTACTEN)*, UPDATE?)>

<!ELEMENT PAKKETTEN (PAKKET+)>
<!ELEMENT PAKKET (PK?, L*, (POLISSEN | WACHTPOLISSEN)*, UPDATE?)>

<!ELEMENT POLISSEN (POLIS+)>
<!ELEMENT POLIS (PK?, L*, (DEKKINGEN | WACHTPOLISSEN | SCHADES| CONTACTEN)*,
UPDATE?)>

<!ELEMENT DEKKINGEN (DEKKING+)>
<!ELEMENT DEKKING (PK?, L*, UPDATE?)>
<!ATTLIST DEKKING ACTIE (opvoeren | wijzigen | verwijderen | identificatie)
#IMPLIED>

<!ELEMENT WACHTPOLISSEN (WACHTPOLIS+)>
<!ELEMENT WACHTPOLIS (PK?, L*, WACHTDEKKINGEN*, UPDATE?)>

<!ELEMENT WACHTDEKKINGEN (WACHTDEKKING+)>
<!ELEMENT WACHTDEKKING (PK?, L*, UPDATE?)>

<!ELEMENT SCHADES (SCHADE+)>
<!ELEMENT SCHADE (PK?, L*, UPDATE?)>
```

```

<!ELEMENT CONTACTEN (CONTACT+)>
<!ELEMENT CONTACT (PK?, L*)*, UPDATE?>

<!-- PK en labels -->
<!ELEMENT PK (#PCDATA)>
<!ELEMENT L (#PCDATA)>
<!ATTLIST L NR WAARDE CDATA #REQUIRED>

<!-- Per entiteit de ADI 2 opties -->
<!ELEMENT UPDATE (DEFAULTS?, MACROS?, MACROS-UITSLUITEN-IN-EXTERNE-ACCEPTATIE?, SLOTACTIES?, VERPLICHTE_VELDEN?, CODES?)>

<!ELEMENT DEFAULTS EMPTY>
<!ATTLIST DEFAULTS UITVOEREN (nee|systeem) 'systeem'>

<!ELEMENT MACROS EMPTY>
<!ATTLIST MACROS UITVOEREN (nee|systeem) 'systeem'>

<!ELEMENT MACROS-UITSLUITEN-IN-EXTERNE-ACCEPTATIE EMPTY>
<!ATTLIST MACROS-UITSLUITEN-IN-EXTERNE-ACCEPTATIE UITVOEREN (ja|systeem) 'systeem'>

<!ELEMENT SLOTACTIES (ADN | FORMULIER)*>
<!ATTLIST SLOTACTIES UITVOEREN (nee|systeem) 'systeem'>

<!ELEMENT VERPLICHTE_VELDEN EMPTY>
<!ATTLIST VERPLICHTE_VELDEN CONTROLLEREN (ja|nee|systeem) 'systeem'>

<!ELEMENT CODES EMPTY>
<!ATTLIST CODES CONTROLLEREN (nee|systeem) 'systeem'>

<!ELEMENT ADN EMPTY>
<!ATTLIST ADN VIEW CDATA #REQUIRED
             VERSIE CDATA #REQUIRED
             UITVOEREN (ja|nee|systeem) 'ja'>

<!ELEMENT FORMULIER EMPTY>
<!ATTLIST FORMULIER SOORT CDATA #REQUIRED
                    NR CDATA #REQUIRED
                    AFDRUKKEN (ja|nee|systeem) 'ja'>

```

**Note:** De acties “wijzigen” en “identificatie” zijn reserved words en hebben (nog) geen sturende functie. Derhalve is het niet toegestaan om deze acties te gebruiken in een XML-string!

### 7.3 Parameters

Naam	Beschrijving
verbose	Optioneel. Indien aanwezig wordt voor elke entiteit de uitgevoerde acties teruggekoppeld.
debug	Optioneel. Indien aanwezig wordt voor elke entiteit de geschreven labels teruggekoppeld.
execute	Optioneel. Indien aanwezig worden de mutaties direct door ADI 2 verwerkt. Het resultaat van de ADI 2 verwerking wordt dan aan het eind van het result element geplaatst. De resulterende records kunnen daarna met WebView opgehaald worden.
view	Verplicht. In dit element wordt de mutatieview meegegeven.
login	Optioneel. Voer de mutatie uit als de opgegeven gebruiker.



## 8 XML-Interface-Muteren Entiteiten

In het entiteitelement worden de wijzigingen doorgegeven:

```
<RELATIE>
  <PK>...</PK>
  <L NR="...">...</L>
  ...
</RELATIE>
```

Met PK wordt de primary key van het record meegegeven zoals deze door XML-Interface-View is aangeleverd. Wanneer dit veld ontbreekt worden de primary key-gegevens uit de labels gehaald. Wanneer de entiteit genest is kunnen deze gegevens ook uit de parent gehaald worden.

**Belangrijk:** met het element <PK> wordt aangegeven of het een bestaande of nieuwe entiteit betreft.

1. Als de <PK> in een entiteit ontbreekt zal deze worden opgevoerd (als nieuw record).
2. Als de <PK> aanwezig is betreft het een mutatie op een bestaande entiteit.
3. Als de <PK> aanwezig is en de entiteit wordt niet gevonden, dan kunnen zich afhankelijk van het soort entiteit twee situaties voordoen:
  1. Bij de soorten Relatie, Polis, Dekking, Wachtpolis of Wachtdekking zal de entiteit worden opgevoerd als nieuw record, respectievelijk in het bestand NAWBES, POLBES, DEKKING, WACBES of WACDEK.
  2. Bij alle andere soorten wordt een foutmelding gegenereerd.

Alleen de te muteren labels dienen te worden meegegeven. Van de meegegeven labels zullen alleen de labels die verschillen met de waarden in ANVA opgenomen worden in de ADI 2-interface.

Wanneer de primary key niet overeenkomt met de corresponderende labels in de view dan zal de XML-Interface-Muteren een fout genereren voor die entiteit. Het bericht zal dan niet verwerkt worden en in de response zal een foutmelding in de teruggekoppelde entiteit gezet worden.

### 8.1 Collecties

In de entiteit kunnen collecties worden genest.

```
<RELATIES>
  <RELATIE>
    <PK>...</PK>
    <L NR="...">...</L>
    ...
    <POLISSEN>
      <POLIS>
        <PK>...</PK>
        <L NR="...">...</L>
        ...
      </POLIS>
    <DEKKINGEN>
      <DEKKING ACTIE="verwijderen">
        <PK>...</PK>
        <L NR="...">...</L>
        ...
      </DEKKING>
    </DEKKINGEN>
    ...
  </POLISSEN>
  ...
</RELATIE>
</RELATIES>
```

Wanneer een entiteit genest is in een entiteit waar deze niet direct toe behoort zal de XML-Interface-Muteren een fout genereren voor die entiteit. Het bericht zal dan niet verwerkt worden en in de response zal een foutmelding in de teruggekoppelde entiteit gezet worden.

## 8.2 Update

Met dit element kunnen een aantal opties per entiteit aangepast worden. De volgende opties zijn beschikbaar:

Element	Attribuut	Waarden	Opmerking
DEFAULTS	UITVOEREN	Nee, systeem	Of de default waarden moeten worden ingevuld
MACROS	UITVOEREN	Nee, systeem	Of de macro's moeten worden uitgevoerd
MACROS-UITSLUITEN- IN-EXTERNE- ACCEPTATIE	UITVOEREN	Ja, systeem	Of de macro's in externe acceptatie moeten worden uitgesloten
SLOTACTIES	UITVOEREN	Nee, systeem	Of de default slotacties moeten worden uitgevoerd
VERPLICHTE_VELDEN	CONTROLLEREN	ja, nee, systeem	Of er gecontroleerd moet worden dat de verplichte velden zijn ingevuld
CODES	CONTROLLEREN	Nee, systeem	Of er gecontroleerd moet worden dat de gewijzigde codevelden een geldige waarde hebben
CODEFILTER	CONTROLLEREN	Nee, systeem	Of er gecontroleerd moet worden dat de gewijzigde codevelden een waarde hebben die wordt toegestaan door het AFD-codefilter voor uniforme polisschermen (volmacht).

### 8.2.1 ADN View

In het element Slotacties kunnen ook nog ADN-views en formulieren opgegeven worden:

```
<SLOTACTIES>
  <ADN VIEW="..." VERSIE="..." UITVOEREN="..." />
  ...
  <FORMULIER SOORT="..." NR="..." AFDRUKKEN="..." />
  ...
</SLOTACTIES>
```

Hiermee kunnen specifieke ADN-berichten en formulieren worden toegevoegd of onderdrukt.

In het ADN-element zijn de VIEW- en VERSIE-attributen verplicht. Het attribuut UITVOEREN kan één van de volgende waarden bevatten: "ja", "nee" of "systeem". De defaultwaarde is "ja".

In het FORMULIER-element zijn de SOORT- en NR-attributen verplicht. Het attribuut AFDRUKKEN kan één van de volgende waarden bevatten: "ja", "nee" of "systeem". De defaultwaarde is "ja".

De volgende tabel geeft aan bij welke entiteiten de slotacties ADN en FORMULIER kunnen worden toegepast:

Element	Toegestaan bij entiteiten
ADN	POLIS en WACHTPOLIS
FORMULIER	RELATIE, POLIS en WACHTPOLIS

## 8.3 Gebruik van ACTIE

Bij een entiteit kan een ACTIE worden opgegeven die aangeeft wat met de entiteit moet gebeuren. Op deze manier kan de standaardwerking van de programmatuur worden bijgestuurd.

De syntax luidt: <ENTITEITNAAM ACTIE="gewenste actie">

Hierbij kan de gewenste actie zijn: "opvoeren" of "verwijderen".

De actie “verwijderen” is alleen beschikbaar voor de entiteit (WACHT)DEKKING en maakt het mogelijk een bestaande dekking te verwijderen.

De actie “opvoeren” is beschikbaar voor de entiteiten RELATIE, (PAKKET/WACHT)POLIS en (WACHT)DEKKING. Hiermee kan worden gestuurd dat bij een opgegeven PK die geen geldige entiteit oplevert een nieuwe entiteit wordt opgevoerd. Hierbij wordt voor het toekennen van een nieuwe PK de bestaande methodiek van de programmatuur gehanteerd. Zonder de actie “opvoeren” zou in dit geval een foutmelding worden gegenereerd.

De acties “wijzigen” en “identificatie” zijn reserved words en hebben (nog) geen sturende functie. Derhalve is het niet toegestaan om deze acties te gebruiken in een XML-string!

## 8.4 Nieuwe Records

Als `XML-Interface-Muteren` het laatste deel van een primary key niet kan vaststellen (b.v. volgnummer van een polis) of het record in ANVA niet aanwezig is gaat de `XML-Interface-Muteren` er van uit dat het om een nieuw record gaat.

Bij een nieuw contract is de maatschappij leidend bij de bepaling of het een polis- of servicecontract betreft; een servicemaatschappij leidt dus automatisch tot een servicecontract.

### Foutmeldingen

Als de `XML-Interface-Muteren` het eerste deel van een primary key niet kan vaststellen (b.v. relatienummer van een polis) zal de `XML-Interface-Muteren` een fout genereren voor die entiteit. Het bericht zal dan niet verwerkt worden en in de response zal een foutmelding in de entiteit gezet worden.

Ook bij polisgegevens in een servicecontract of bij servicegegevens in een poliscontract zal de response een foutmelding in de entiteit bevatten.

## 8.5 Execute

Wanneer de parameter “execute” aanwezig is worden de wijzigingen meteen door de `XML-Interface` doorgevoerd. Wanneer hier een fout optreedt wordt deze in het `EXECUTE`-element van de uitvoer teruggegeven. Hierbij worden zo mogelijk ook de entiteit en het label aangeduid waar de fout optrad.

Tijdens het uitvoeren van de wijzigingen wordt er geen lijst gegenereerd.

## 8.6 Result

In het resultaat wordt in de `XML-Interface` “Transactie ID” aangegeven waaronder de wijzigingen zijn weggeschreven. Ook wordt de aangeleverde view zonder labels teruggekoppeld. Waar bekend zijn de primaire sleutels aangegeven. Als er voor de entiteit gewijzigde labels zijn aangeleverd dan wordt ook een transactievolgnummer teruggekoppeld.

Als de wijzigingen meteen doorgevoerd worden wordt er ook een `EXECUTE`-element teruggekoppeld.

Het `RESULT`-element zal in de meeste gevallen de `CODE` “0” aangeven. Dit betekent echter alleen dat alles goed is gegaan totdat de view verwerkt werd. Om te controleren of de hele verwerking gelukt is zal ook gecontroleerd moeten worden of er niet ergens een `ERROR`-element is teruggegeven.

Als er een fout is opgetreden of als de wijzigingen meteen worden doorgevoerd zal de transactie automatisch uit het systeem verwijderd worden.

### 8.6.1 DTD Result

De DTD voor Result is als volgt:

```
<?xml version="1.0" encoding="ibm850"?>
<!ELEMENT RESPONSE (RESULT)>

<!ELEMENT RESULT (CREATING_TRANSACTION, TRANSACTION_ID,
                 (RELATIES | PAKKETTEN | POLISSEN | DEKKINGEN
                  | WACHTPOLISSEN | WACHTDEKKINGEN | SCHADES | CONTACTEN)*,
                 EXECUTE?, DELETING_TRANSACTION?)>

<!ELEMENT CREATING_TRANSACTION EMPTY> <!-- Indien verbose -->
<!ELEMENT TRANSACTION_ID (#PCDATA)>

<!-- View -->
<!ELEMENT RELATIES (RELATIE+)>
<!ELEMENT RELATIE (PK?, L*, (ERROR | ((TRANSACTION_VOLGNR, PK?)?,
                                     (PAKKETTEN | POLISSEN | WACHTPOLISSEN | CONTACTEN)*,
                                     UPDATE?))))>

<!ELEMENT PAKKETTEN (PAKKET+)>
<!ELEMENT PAKKET (PK?, L*, (ERROR | ((TRANSACTION_VOLGNR, PK?)?,
                                     (POLISSEN | WACHTPOLISSEN)*, UPDATE?))))>
```

```

<!ELEMENT POLISSEN (POLIS+)>
<!ELEMENT POLIS (PK?, L*, (ERROR | ((TRANSACTION VOLGNR, PK?)?,
    (DEKKINGEN | WACHTPOLISSEN | SCHADES | CONTACTEN)*,
    UPDATE?)))>

<!ELEMENT DEKKINGEN (DEKKING+)>
<!ELEMENT DEKKING (PK?, L*, (ERROR | ((TRANSACTION VOLGNR, PK?)?, UPDATE?)))>

<!ELEMENT WACHTPOLISSEN (WACHTPOLIS+)>
<!ELEMENT WACHTPOLIS (PK?, L*, (ERROR | ((TRANSACTION_VOLGNR, PK?)?,
    WACHTDEKKINGEN*, UPDATE?)))>

<!ELEMENT WACHTDEKKINGEN (WACHTDEKKING+)>
<!ELEMENT WACHTDEKKING (PK?, L*, (ERROR | ((TRANSACTION VOLGNR, PK?)?,
    UPDATE?)))>

<!ELEMENT SCHADES (SCHADE+)>
<!ELEMENT SCHADE (PK?, L*, (ERROR | ((TRANSACTION_VOLGNR, PK?)?, UPDATE?)))>

<!ELEMENT CONTACTEN (CONTACT+)>
<!ELEMENT CONTACT (PK?, L*, (ERROR | ((TRANSACTION VOLGNR, PK?)?, UPDATE?)))>

<!-- PK en labels -->
<!ELEMENT PK (#PCDATA)>
<!ELEMENT L (#PCDATA)> <!-- Indien debug -->
<!ATTLIST L NR CDATA #REQUIRED>

<!-- Per entiteit de ADI 2 opties -->
<!ELEMENT UPDATE (DEFAULTS?, MACROS?, MACROS-UITSLUITEN-IN-EXTERNE-
ACCEPTATIE?, SLOTACTIES?, VERPLICHTE VELDEN?, CODES?)> <!-- Indien debug -->

<!ELEMENT DEFAULTS EMPTY>
<!ATTLIST DEFAULTS UITVOEREN (nee|systeem) #REQUIRED>

<!ELEMENT MACROS EMPTY>
<!ATTLIST MACROS UITVOEREN (nee|systeem) #REQUIRED>

<!ELEMENT MACROS-UITSLUITEN-IN-EXTERNE-ACCEPTATIE EMPTY>
<!ATTLIST MACROS-UITSLUITEN-IN-EXTERNE-ACCEPTATIE UITVOEREN (ja|systeem)
#REQUIRED>

<!ELEMENT SLOTACTIES (ADN | FORMULIER)*>
<!ATTLIST SLOTACTIES UITVOEREN (nee|systeem) #REQUIRED>

<!ELEMENT VERPLICHTE_VELDEN EMPTY>
<!ATTLIST VERPLICHTE_VELDEN CONTROLEREN (ja|nee|systeem) #REQUIRED>

<!ELEMENT CODES EMPTY>
<!ATTLIST CODES CONTROLEREN (nee|systeem) #REQUIRED>

<!ELEMENT ADN EMPTY>
<!ATTLIST ADN VIEW CDATA #REQUIRED
    VERSIE CDATA #REQUIRED
    UITVOEREN (ja|nee|systeem) #REQUIRED>

<!ELEMENT FORMULIER EMPTY>
<!ATTLIST FORMULIER SOORT CDATA #REQUIRED
    NR CDATA #REQUIRED
    AFDRUKKEN (ja|nee|systeem) #REQUIRED>

<!-- Error -->
<!ELEMENT ERROR (DESCRIPTION, CODE LOCATION, TRANSACTION VOLGNR?, L?)>
<!ATTLIST ERROR CODE CDATA #REQUIRED>

<!ELEMENT EXECUTE (RELATIE | PAKKET | POLIS | DEKKING
    | WACHTPOLIS | WACHTDEKKING | SCHADE | CONTACTEN)*>
    <!-- Terugkoppeling van PK -->

<!ELEMENT DELETING_TRANSACTION EMPTY> <!-- Indien verbose -->

```

## 9 XML-Interface voorbeelden

### 9.1 Mutaties in de Wacht

**Notitie** Een niet-actieve collectiviteit kan ingeschoten worden in een polis in de wacht, daarbij worden ook kortingen toegepast.

#### 9.1.1 Wachtmutatie op bestaande polis

Bij de WACHTPOLIS moet in het PK-element de primary key van de wachtpolis meegegeven worden. Deze bestaat uit het relatiefolnummer van de desbetreffende polis met daarachter het volgnummer van de wacht. Wordt een wacht-volgnummer niet, leeg of nul meegegeven dan wordt dit beschouwd als een nieuw aan te maken wacht. Op basis van het relatiefolnummer wordt in Anva de bestaande polis gezocht, wordt deze gevonden dan wordt een wachtsituatie aangemaakt bij deze bestaande polis. Is nog geen polis aanwezig met het betreffende relatiefolnummer dan wordt een nieuwe polis in de wacht gemaakt.

```
<RELATIES>
  <RELATIE>
    <PK>21000033</PK>
    <L NR="...">...</L>
    ...
    <WACHTPOLISSEN>
      <WACHTPOLIS>
        <PK>WACBES_210000330200102001000</PK>
        <L NR="...">...</L>
        ...
      </WACHTPOLIS>
    </WACHTPOLISSEN>
    ...
  </RELATIE>
</RELATIES>
```

#### 9.1.2 Mutatie van bestaande dekking

Bij de WACHTDEKKING moet in het PK-element de primary key van de wachtdekking meegegeven worden. Deze bestaat uit het relatiefolnummer van de desbetreffende polis aangevuld met het wachtvolnummer en het volgnummer van de dekking. Het wachtvolnummer moet gelijk zijn aan die op polisniveau. Wordt een wacht-volgnummer leeg, nul of niet meegegeven dan wordt dit beschouwd als een nieuw aan te maken wacht.

Op basis van de combinatie relatiefolnummer en volgnummer dekking wordt in Anva de bestaande dekking gezocht, wordt deze gevonden dan wordt een wachtsituatie aangemaakt bij deze bestaande dekking.

```
<RELATIES>
  <RELATIE>
    <PK>21000033</PK>
    <L NR="...">...</L>
    ...
    <WACHTPOLISSEN>
      <WACHTPOLIS>
        <PK>WACBES_210000330200102001000</PK>
        <L NR="...">...</L>
        ...
      </WACHTPOLIS>
      <WACHTDEKKINGEN>
        <WACHTDEKKING>
          <PK>WACDEK_21000033020010200100000001</PK>
          <L NR="...">...</L>
          ...
        </WACHTDEKKING>
      </WACHTDEKKINGEN>
    </WACHTPOLISSEN>
    ...
  </RELATIE>
</RELATIES>
```

## 9.2 Verwijderen Dekking

### 9.2.1 Verwijderen Dekking

Alleen een het geval van een dekking is het mogelijk om deze te verwijderen. Dit is mogelijk door de actie “verwijderen” mee te geven aan de dekking.

```

<RELATIES>
  <RELATIE>
    <PK>...</PK>
    <L NR="...">...</L>
    ...
    <POLISSEN>
      <POLIS>
        <PK>...</PK>
        <L NR="...">...</L>
        <L NR="10010">20031231</L>
        ...
        <UPDATE>
          <SLOTACTIES UITVOEREN="systeem"/>
        </UPDATE>
        <DEKKINGEN>
          <DEKKING ACTIE="verwijderen">
            <PK>...</PK>
            <L NR="...">...</L>
            ...
          </DEKKING>
          ...
        </DEKKINGEN>
        ...
      </POLIS>
    </POLISSEN>
    ...
  </RELATIE>
</RELATIES>

```

Bij verwijderen van een dekking is het belangrijk dat op polisniveau wordt aangegeven dat de slotacties worden uitgevoerd. Alleen dan wordt namelijk de bijhorende premieverrekening uitgevoerd.

**Let op** Bij bepaling van de periode waarover deze verrekening plaats zal vinden wordt uitgegaan van de wijzigingsdatum in de polis. Het is dus van belang dat deze wijzigingsdatum meegegeven wordt in het bericht (label 10010 in de polis).

### 9.2.2 Dekking verwijderen in de Wacht

In de wacht verwijderen van een dekking gebeurt op dezelfde wijze als mutatie van een dekking in de wacht. Een dekking kan verwijderd worden door "actie=verwijderen" toe te voegen aan het element <WACHTDEKKING>. E.e.a. als in onderstaand voorbeeld.

```

<RELATIES>
  <RELATIE>
    <PK>21000033</PK>
    <L NR="...">...</L>
    ...
    <WACHTPOLISSEN>
      <WACHTPOLIS>
        <PK>WACBES_210000330200102001000</PK>
        <L NR="...">...</L>
        ...
      </WACHTPOLIS>
      <WACHTDEKKINGEN>
        <WACHTDEKKING ACTIE="verwijderen">
          <PK>WACDEK_21000033020010200100000001</PK>
          <L NR="...">...</L>
          ...
        </WACHTDEKKING>
        ...
      </WACHTDEKKINGEN>
    </WACHTPOLISSEN>
    ...
  </RELATIE>
</RELATIES>

```



### 9.3 Mandaatgegevens (SEPA)

**Let op** Een mandaat-request kan niet genest worden binnen een relatie- of polis-request. Ze kunnen wel achter elkaar als twee aparte requests in één XML. In het geval van een nieuwe relatie is de volgorde: eerst een request voor relatie en daarna een request voor het mandaat. In deze volgorde worden **beide requests** verwerkt, maar er wordt slechts **één response** gegeven.

Om toch voor beide requests een response te krijgen, moet u één request per XML aanmaken. Dus in het geval van een nieuwe relatie: eerst een XML met een request voor de relatie en daarna een XML met een request voor het mandaat. Zo krijgt elke XML-request een aparte response.

#### 9.3.1 Mandaatgegevens opvragen

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="oms"/>
    <PARAM NAME="display"/>
    <PARAM NAME="select">
      <MANDATEN>
        <MANDAAT>
          <L NR="99318"/>
          <L NR="99319"/>
          <L NR="99320"/>
          <L NR="99321"/>
          <L NR="99324"/>
          <L NR="99325"/>
          <L NR="99330"/>
        </MANDAAT>
        <WHERE>
          <PK EQ="MANDAAT_00010NL02INGB0000186030"/>
        </WHERE>
      </MANDATEN>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

#### 9.3.2 Mandaatgegevens opvragen zonder kantoorcode (geen concern of concern met één boekhouding)

In dit geval gebruikt u kantoorcode '00000'.

```
<REQUEST>
  <FUNCTION NAME="webview.select">
    <PARAM NAME="oms"/>
    <PARAM NAME="display"/>
    <PARAM NAME="select">
      <MANDATEN>
        <MANDAAT>
          <L NR="99318"/>
          <L NR="99319"/>
          <L NR="99320"/>
          <L NR="99321"/>
          <L NR="99324"/>
          <L NR="99325"/>
          <L NR="99330"/>
        </MANDAAT>
        <WHERE>
          <PK EQ="MANDAAT_00000NL02INGB0000186030"/>
        </WHERE>
      </MANDATEN>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

### 9.3.3 Mandaatgegevens muteren

Bij het muteren van mandaatgegevens dient de primary key (kantoorcode en IBAN) te worden opgegeven. Label 99321 (laatst gebruikt) kan niet worden gemuteerd.

```
<REQUEST>
  <FUNCTION NAME="webmuteren.update">
    <PARAM NAME="execute"/>
    <PARAM NAME="view">
      <MANDATEN>
        <MANDAAT>
          <PK>MANDAAT_0000010NL02INGB0000186030</PK>
          <L NR="99318">20130524</L>
          <L NR="99319">20140524</L>
          <L NR="99320">RCUR</L>
          <L NR="99324"/>Janssen</L>
          <L NR="99325"/>Postweg</L>
          <L NR="99330"/>Kijkduin</L>
        </MANDAAT>
      </MANDATEN>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

### 9.3.4 Mandaatgegevens muteren zonder kantoorcode (geen concern of concern met één boekhouding)

In dit geval gebruikt u kantoorcode '00000'.

```
<REQUEST>
  <FUNCTION NAME="webmuteren.update">
    <PARAM NAME="execute"/>
    <PARAM NAME="view">
      <MANDATEN>
        <MANDAAT>
          <PK>MANDAAT_00000NL02INGB0000186030</PK>
          <L NR="99318">20130524</L>
          <L NR="99319">20140524</L>
          <L NR="99320">RCUR</L>
        </MANDAAT>
      </MANDATEN>
    </PARAM>
  </FUNCTION>
</REQUEST>
```

## 9.4 Nieuwe relatie opvoeren

ANVA maakt een relatie aan indien er geen relatienummer wordt meegegeven. Label **40 Soort relatie** is verplicht.

```
<REQUEST>
  <FUNCTION NAME="webmuteren.update">
    <PARAM NAME="execute"/>
    <PARAM NAME="view">
      <RELATIES>
        <RELATIE>
          <L NR="40">1</L> | Soort relatie
          <L NR="27">2</L> | Wijzigingsreden relatie
          <L NR="2">Jansen</L> | Naam relatie
          <L NR="3">J.J.</L> | voorletters relatie
          <L NR="58">20130729</L> Wijzigingsdatum
        <UPDATE>
        </UPDATE>
      </RELATIE>
    </RELATIES>
  </PARAM>
</FUNCTION>
</REQUEST>
```

### 9.4.1 Object opvoeren bij bestaande relatie

```
<REQUEST>
  <FUNCTION NAME="webmuteren.update">
    <PARAM NAME="execute"/>
    <PARAM NAME="view">
      <RELATIES>
        <RELATIE>
          <PK>NAWBES_00000001</PK> | Id van de relatie
          <OBJECTEN>
            <OBJECT>
              <L NR="51000">MA</L> | Objectsoort
              <L NR="51001">Object Machine via XML</L>
              <L NR="51004">1</L>
              <L NR="51005">7</L>
              <L NR="51006">J</L>
              <L NR="51007">Samsung</L>
              <L NR="51008">468546</L>
              <L NR="51009">2014</L>
              <L NR="51010">15000</L>
              <L NR="51011">N</L>
              <L NR="51012">J</L>
              <L NR="51013">D</L>
              <L NR="51014">J</L>
              <L NR="51015">20000</L>
              <L NR="51016">50000</L>
            </OBJECT>
          </OBJECTEN>
        </RELATIE>
      </RELATIES>
    </PARAM>
```

### 9.4.2 Bestand object bij een relatie muteren

```
<REQUEST>
  <FUNCTION NAME="webmuteren.update">
    <PARAM NAME="execute"/>
    <PARAM NAME="view">
      <RELATIES>
        <RELATIE>
          <PK>NAWBES_00000001</PK>
```

```
<OBJECTEN>
  <OBJECT>
    <PK>OBJECT_000000012b6f8a1e-a17f-4860-9e2c-28ffae2ca850</PK>
    <L NR="51001">Object Machine via XML Mutatie</L>
    <L NR="51004">2</L>
    <L NR="51005">5</L>
    <L NR="51006">N</L>
    <L NR="51011">J</L>
    <L NR="51012">J</L>
    <L NR="51013">W</L>
    <L NR="51014">J</L>
  </OBJECT>
</OBJECTEN>
</RELATIE>
</RELATIES>
</PARAM>
```

### 9.4.3 Compliancy-check uitvoeren

De Compliancy-check kan worden uitgevoerd bij het inschieten van een XML. Voorwaarde hiervoor is, dat de volgende opties in pad **BYSS** op **Ja** moeten staan:

- Compliancy Check.
- Automatische bevraging Compliancy Check.

Op niveau <RELATIE> moet de volgende tag aanwezig zijn:

```
<SLOTACTIES UITVOEREN="Systeem" />
```

## 9.5 Verwijzing opgeven

Bedoeling is om in polis 2001 van relatie 1 een verwijzing (polis wordt betaald door) op te nemen naar relatie 2.

```

<REQUEST>
  <FUNCTION NAME="webmuteren.Update">
    <PARAM NAME="execute"/>
    <PARAM NAME="view">
      <RELATIES>
        <RELATIE>
          <PK>NAWBES_00000001</PK>
          <UPDATE>
            <CODES CONTROLEREN="Nee"/>
            <VERPLICHTE_VELDEN CONTROLEREN="Nee"/>
            <MACROS UITVOEREN="systeem" />
            <MACROS-UITSLUITEN-IN-EXTERNE-ACCEPTATIE UITVOEREN="systeem" />
          </UPDATE>
          <POLISSEN>
            <POLIS>
              <PK>POLBES_000000010200102001</PK>
              <L NR="10041">34368</L>
              <UPDATE>
                <CODES CONTROLEREN="Nee"/>
                <VERPLICHTE_VELDEN CONTROLEREN="Nee"/>
                <SLOTACTIES UITVOEREN="Systeem" />
              </UPDATE>
              <VERWIJZINGEN>
                <VERWIJZING>
                  <L NR="1">2</L>
                  <L NR="99094">2</L>
                  <UPDATE>
                    <CODES CONTROLEREN="nee" />
                    <DEFAULTS UITVOEREN="systeem" />
                    <MACROS UITVOEREN="systeem" />
                    <MACROS-UITSLUITEN-IN-EXTERNE-ACCEPTATIE UITVOEREN="systeem" />
                    <SLOTACTIES UITVOEREN="nee" />
                    <VERPLICHTE_VELDEN CONTROLEREN="nee" />
                  </UPDATE>
                </VERWIJZING>
              </VERWIJZINGEN>
            </POLIS>
          </POLISSEN>
        </RELATIE>
      </RELATIES>
    </PARAM>
  </FUNCTION>
</REQUEST>

```

## 9.6 Formulier gegeneren bij muteren of invoeren van een (wacht)polis

Dit voorbeeld is van het opgeven van een formulier in de slotacties bij het toevoegen of muteren van een polis (in de wacht).

```
<UPDATE>
  <SLOTACTIES UITVOEREN="Nee">
    <FORMULIER SOORT="AB001" NR="BB001" AFDrukKEN="DIRECT" RETURN-DOSSIERNR="JA"/>
  </SLOTACTIES>
  <VERPLICHTE_VELDEN CONTROLEEREN="Nee"/>
  <DEFAULTS UITVOEREN="Nee"/>
  <CODES CONTROLEEREN="Nee"/>
  <MACROS UITVOEREN="Nee"/>
</UPDATE>
```