

2013-02-08

# **SHS Version 2.0 SOAP-based Protocol Riktlinjer för Domänschema**

**Försäkringskassan - Swedish Social Insurance Agency**

Utgåva PA3  
2013-02-08

Copyright © 2012, 2013 Swedish Social Insurance Agency. All Rights Reserved.



2013-02-08

## Utgåvehistorik

<i>Utgåva</i>	<i>Revision Datum</i>	<i>Beskrivning</i>	<i>Ändringarna gjorda av</i>	<i>Definitiv revision fastställd av</i>
PA1	2012-04-18	Dokumentet skapat baserat på RIV TA Domänschema 2.1	<a href="mailto:Anders.asplund@callistaenterprise.se">Anders.asplund@callistaenterprise.se</a>	
PA2	2012-05-02	Återinfört Uppdrag-Resultat och Informationsspridning	<a href="mailto:Anders.asplund@callistaenterprise.se">Anders.asplund@callistaenterprise.se</a>	
	2012-10-12	SHS Version 2.0, efter 1'a remiss		
PA3	2013-02-08	Slutgiltig version SHS 2.0		

2013-02-08

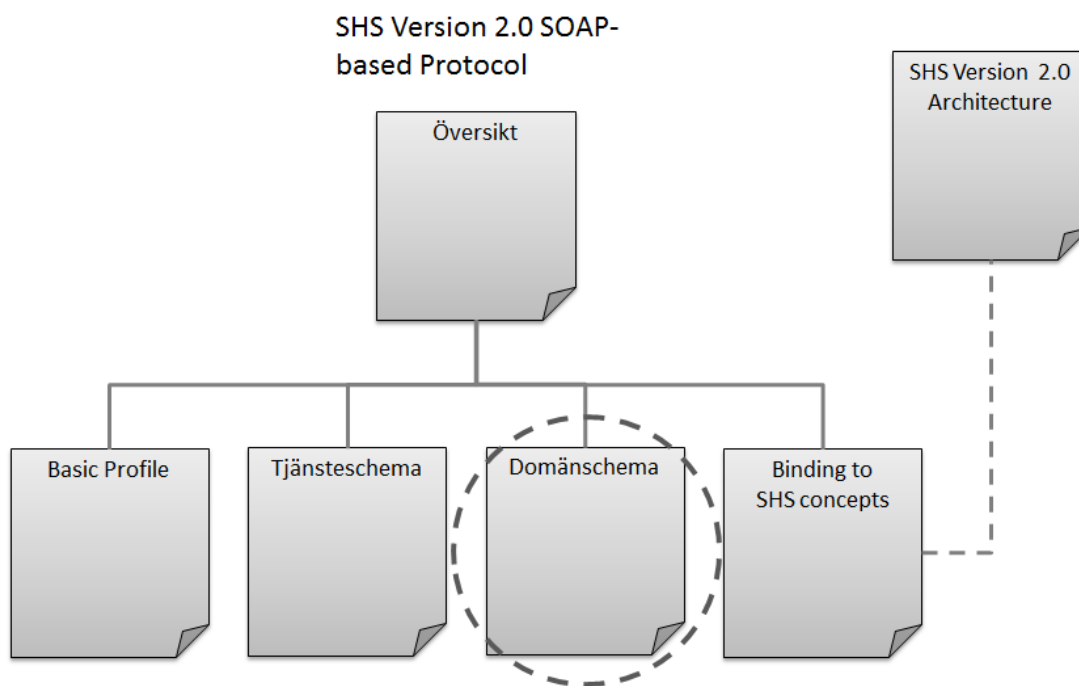
**INNEHÅLLSFÖRTECKNING**

<b>1. INLEDNING</b>	<b>4</b>
1.1. MÅLGRUPP	4
1.2. SYFTE	4
1.3. TILLGÅNGLIGHET	4
1.4. UPPHOVSMAN	4
1.5. REFERENSER	4
<b>2. MEDDELANDEREGLER</b>	<b>6</b>
REGEL #1, DESIGNMÖNSTER FÖR DOMÄNSCHEMAN	6
REGEL #2, NAMN PÅ XSD-FILEN	6
REGEL #3, NAMN PÅ TARGET NAMESPACE	6
REGEL #4, ANVÄNDNING AV SCHEMA-ATTRIBUTET VERSION	6
REGEL #5, ANVÄNDNING AV ANY-ELEMENT FÖR UTÖKNINGSBARHET	6
REGEL #9, BAKÅTKOMPATIBLA ELEMENT I UTÖKNINGSSCHEMA	6
REGEL #6 NATIONELLA TECKEN	8

2013-02-08

## 1. Inledning

Detta dokument beskriver regelverket SHS Version 2.0 SOAP-based Protocol Domänschema.



### 1.1. Målgrupp

Denna anvisning riktar sig till dem som ska specificera XML-scheman för tjänstekontrakt i en nationell tjänstedomän. Anvisningen innehåller endast regeluppsättningen. För bakgrund, motiv, krav samt de principer som ligger till grund för framtagning av reglerna hänvisas till SHS Version 2.0 SOAP-based Protocol Översikt [R1].

### 1.2. Syfte

Syftet med denna anvisning är att beskriva designregler och namngivningsregler för de meddelandetyper som skall användas inom en tjänstedomän.

### 1.3. Tillgänglighet

Specifikationerna för SHS Version 2.0 publiceras på Försäkringskassans hemsida [www.fk.se](http://www.fk.se).

### 1.4. Upphovsman

SHS Version 2.0 SOAP-based Protocol är framtagen av Försäkringskassan och baserad på RIV TA version 2.1 med upphovsman SKL(Sveriges Kommuner och Landsting).

### 1.5. Referenser

Ref	Dokument	Beskrivning och ev. webbadress	Ansvarig
[R1]	Översikt SHS Version 2.0 SOAP-based Protocol	Bakgrund, motiv, krav samt de principer som ligger till grund för utvecklingen av denna anvisning.	Försäkringskassan



2013-02-08

<b>[R2]</b>	SHS Version 2.0 SOAP-based Protocol Basic Profile	Exempel på anvisning för profil som pekar ut användningen av denna anvisning för specifikation av meddelandehåll (teknisk del).	Försäkringskassan
<b>[R3]</b>	Beskrivning av "Venetian Blind"	Dokumentet beskriver det designmönster som tillämpas för XML Schema design i denna anvisning. Webblänk till hemsidan: <a href="http://www.xfront.com/GlobalVersusLocal.html#ThirdDesign">http://www.xfront.com/GlobalVersusLocal.html#ThirdDesign</a>	Okänd.
<b>[R4]</b>	W3C-rapport om utökningsbara XML-scheman	Beskriver problemställningar och strategier för design av meddelanden som ger bra stöd för versionshantering. Versioneringsstrategin som beskrivs i denna översikt och som tillämpas i RIV Teknisk Anvisning Tjänsteschema är baserad på strategi nr 2.5 i denna rapport. Webblänk till rapportens hemsida: <a href="http://www.w3.org/2001/tag/doc/versioning-xml">http://www.w3.org/2001/tag/doc/versioning-xml</a>	W3C
<b>[R5]</b>	WS-I Basic Profile	" Defines the WS-I Basic Profile 1.1, consisting of a set of non-proprietary Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability " Webblänk till WS-I Basic Profile: <a href="http://www.ws-i.org/Profiles/BasicProfile-1.1.html">http://www.ws-i.org/Profiles/BasicProfile-1.1.html</a>	The Web Services Interoperability Organization och ISO



2013-02-08

## 2. Meddelanderegler

### **Regel #1, designmönster för domänscheman**

"Venetian Blind" [R3] *skall* användas som designmönster för domänscheman. Designmönstret Venetian Blind innebär följande:

- Den interna strukturen i ett meddelande byggs upp med hjälp av globalt deklarerade typer. Med "globalt deklarerade" menas deklarerationer som görs direkt under schema-elementet.
- Endast rotelementet är deklarerat som ett globalt element. I web-service-fallet innebär det att request- och response-elementen är globalt deklarerade element medan resten är typer.

*Anm. I vissa fall kan även andra element än request och response behöva vara globala, t ex används element-referenser till globala element i importerade scheman för att stödja versioneringsstrategin, se nedan.*

**Motiv:** Interoperabilitet, WS-I Basic Profile

### **Regel #2, namn på xsd-filen**

Schema-filen för ett domänschema *skall* namnges enligt följande regel: \${tjänstedomän}\_\${m.n}.xsd

**Motiv:** Att ha med versionsnummer i namnet på källkodsfiler är generellt sett något man försöker undvika då det försvårar användning av versionshanteringsverktyg (t ex Subversion, Microsoft Visual Studio). I fallet med domänschema behöver man dock kunna hantera flera olika versioner samtidigt (i byggsystem mm) och för att underlätta den hanteringen ingår versionsnumret i filnamnet på tjänstekontrakt och tjänstescheman.

**Exempel:** itintegration\_monitoring\_1.0.xsd

### **Regel #3, namn på target namespace**

Attributet targetNamespace på schema-elementet *skall* ha ett värde som definieras av följande regel:

urn:shs:\${tjänsteDomän}:\${m}

**Motiv:** Användningen av major-version i namnrymden är en av att följa fastslagen versioneringsstrategi [R4]. Att ha en unik namnrymd per tjänstekontrakt (tjänsteinteraktion + roll) är en förutsättning för att följa WS-I Basic Profiles [R5] regel om "operation signature". Det också generellt goda förutsättningar för att implementera generella bryggor och tjänsteväxlar

**Exempel:** urn:shs:itintegration:monitoring:1

### **Regel #4, användning av schema-attributet version**

Schema-attributet version *bör* sättas till "m.n"

**Motiv:** Då namnrymden inte innehåller minor-version, ger detta en dokumentation som följer intentionen med attributet.

**Exempel:** <schema ... version="1.0">

### **Regel #5, användning av any-element för utökningsbarhet**

För att uppnå framåtkompatibilitet *skall* ett xsd:any element läggas in sist i alla komplexa typer som ska kunna utökas, exempel:

**Motiv:** För att uppnå framåtkompatibilitet måste man "förbereda" sina XML scheman för framtida utökningsbarhet. Detta är en del av den tillämpade strategin för versionering [R5].

**Exempel:**

```
<xs:complexType name="SomeType">
  <xs:sequence>
    <xs:element name="someElement" type="xsd:string" />
    <xs:element name="someOtherElement" type="xsd:int" />
    <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" namespace="##other"/>
  </xs:sequence>
</xs:complexType>
```

### **Regel #9, bakåtkompatibla element i utökningsschema**

2013-02-08

För att skapa en ny minor-version av ett tjänsteschema, *skall* följande regler följas:

- De nya icke-obligatoriska elementen läggs till i befintligt schema närmast före any-elementet i den komplexa typ som ska utökas. Dessa nya element har ingen typ, utan refererar (xsd:ref="...") element som är rotelement i en ny schema-fil (*utöknings-schema*)
- Definieras de nya icke-obligatoriska elementen i en ny schema-fil (utökningschema) med ett namn som följer följande regel: \${tjänstedomän}\_\${m.n}\_ext.xsd
- Utöknings-schemat ska ha en targetNamespace enligt följande regel: urn:shs:\${tjänsteDomän}:\${m.n}
- Domänschemat importerar (xsd:import) utöknings-schemat som ges namnrymdsalias enligt följande regel: m\${n}
- Domänschemats versionsattribut ändras till den nya minor-versionen.
- I nästa major-version av tjänsteschemat flyttas element-deklarationerna in från alla utöknings-scheman (det finns ett för varje minor-version som tillkommit sedan förra major-versionen skapades).

**Motiv:** Detta förfarande är en konsekvens av vald strategi för versionering [R5]. Se [R4] för ytterligare bakgrund.

**Exempel:** Domänschema, ny minorversion: **crm\_scheduling\_1.1.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="urn:shs:crm:scheduling:1" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:m1="urn:shs:crm:scheduling:1.1" targetNamespace="urn:shs:crm:scheduling:1"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.1">
  <xs:import namespace="urn:shs:crm:scheduling:1.1" schemaLocation="crm_scheduling_1.1_ext.xsd"/>
  <xs:complexType name="SubjectOfCareType">
    <xs:sequence>
      <xs:element name="phone" type="xs:string" minOccurs="0"/>
      <xs:element name="email" type="xs:string" minOccurs="0"/>
      <xs:element name="address" type="xs:string" minOccurs="0"/>
      <xs:element name="coaddress" type="xs:string" minOccurs="0"/>
      <xs:element ref="m1:firstName" minOccurs="1"/>
      <xs:element ref="m1:middleName" minOccurs="1"/>
      <xs:element ref="m1:lastName" minOccurs="1"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Utöknings-schema med element som tillkommit i 1.1: **crm\_scheduling\_1.1\_ext.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="urn:shs:crm:scheduling:1.1" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:shs:crm:scheduling:1.1" elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.1">
  <xs:element name="firstName" type="xs:string"/>
  <xs:element name="middleName" type="xs:string"/>
  <xs:element name="lastName" type="xs:string"/>
</xs:schema>
```

Vid nästa major-version (i detta exempel med ett nytt obligatoriskt element) integreras elementen från mellanliggande utöknings-scheman i huvudschemat: **crm\_scheduling\_2.0.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="urn:shs:crm:scheduling:2" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:m1="urn:shs:crm:scheduling:1.1" targetNamespace="urn:shs:crm:scheduling:2"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0">
  <xs:complexType name="SubjectOfCareType">
    <xs:sequence>
      <xs:element name="phone" type="xs:string" minOccurs="0"/>
      <xs:element name="email" type="xs:string" minOccurs="0"/>
      <xs:element name="address" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```



2013-02-08

```
<xs:element name="coaddress" type="xs:string" minOccurs="0"/>
<xs:element name="firstName" type="xs:string" minOccurs="1"/>
<xs:element name="middleName" type="xs:string" minOccurs="1"/>
<xs:element name="lastName" type="xs:string" minOccurs="1"/>
<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

</xs:schema>
```

## Regel #6 Nationella tecken

Domänscheman ska undvika att använda nationella tecken i såväl elementnamn, attributnamn som vid listning av värdemängder för uppräknings typer. Följande exempel bör därför undvikas:

```
<xs:simpleType name="Å">
  <xs:annotation>
    <xs:documentation>...</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Återställas helt" />
    <xs:enumeration value="Återställas delvis" />
    <xs:enumeration value="Det går inte att bedöma" />
  </xs:restriction>
</xs:simpleType>
```

**Motiv:** För att undvika interoperabilitetsproblem bör man ej använda sig av nationella tecken när man definierar typer som kommer att användas i ett tjänstekontrakt. Ofta uppstår annars fel vid kodgenerering från schemat.