

Sammanfattad basjänst

för grundläggande uppgifter om företag

Teknisk guide gemensamt tekniskt gränssnitt

Version 4.0

Innehållsförteckning

1	Ändringshistorik.....	4
2	Inledning.....	5
3	Tekniskt gränssnitt.....	5
3.1	Gemensamma XML-scheman.....	5
3.1.1	XML-schema informationsutbytesmodell företag.....	5
3.1.1.1	Aktier.....	6
3.1.1.2	AntalAnstalldaStorleksklass.....	7
3.1.1.3	AntalAnstalldaStorleksklassArbetsstalle.....	8
3.1.1.4	Arbetsgivarregistrerad.....	9
3.1.1.5	ArbetsstalleBenamning.....	9
3.1.1.6	ArsomsättningStorleksklass.....	9
3.1.1.7	Avvecklingsförfarande.....	11
3.1.1.8	Belagenhetsadress.....	13
3.1.1.9	CFARNummer.....	13
3.1.1.10	EPostAdress.....	13
3.1.1.11	Firmatecknare.....	13
3.1.1.12	FirmateckningBeskrivning.....	13
3.1.1.13	Fodelsedatum.....	14
3.1.1.14	ForetagAvregistreratDatum.....	14
3.1.1.15	ForetagRegistreratDatum.....	14
3.1.1.16	ForetagNamn.....	14
3.1.1.17	Foretagsform.....	14
3.1.1.18	Funktionar.....	15
3.1.1.19	FysiskPersonNamn.....	17
3.1.1.20	Juridisk form.....	18
3.1.1.21	Kommun.....	19
3.1.1.22	Lan.....	19
3.1.1.23	Momsregistrerad.....	19
3.1.1.24	Namnskydd.....	20
3.1.1.25	PersonIdentitetsbeteckning.....	20
3.1.1.26	Postadress.....	21
3.1.1.27	Rakensapsar.....	21
3.1.1.28	Skatteform.....	22
3.1.1.29	SNI.....	22
3.1.1.30	Telefonnummer.....	22
3.1.1.31	Verksamhetsbeskrivning.....	22
3.1.2	XML-schema metadata.....	23
3.1.2.1	Anvandare.....	23
3.1.2.2	Anvandningsomrade.....	24
3.1.2.3	Datakonsument.....	24
3.1.2.4	Dataproducent.....	25
3.1.2.5	Formedlare.....	25
3.1.2.6	JuridiskForm.....	26
3.1.2.7	MeddelandeId.....	26
3.1.2.8	PartId.....	26

3.1.2.9	Service.....	26
3.1.2.10	Tidstempel.....	27
3.1.2.11	TransaktionId	27
3.1.2.12	TTL.....	27
3.1.3	XML-schema fel	27
3.1.3.1	Fel och feltyp.....	28
3.1.3.2	FelKod och FelBeskrivning.....	30
3.1.4	XML-schema rollintyg	30
3.1.4.1	RollIntyg.....	30
3.1.4.2	RollIntygMetadata.....	31
3.1.4.3	RollIntygId.....	31
3.1.4.4	RollIntygUtfardat.....	31
3.1.4.5	RollIntygUtfardatAv.....	31
3.1.4.6	RollIntygUtfardatTill.....	32
3.1.4.7	RollIntygUtfardatForPerson	32
3.1.4.8	RollIntygUtfardatForForetag	32
3.1.4.9	Roll.....	33
3.2	Felhantering.....	34
3.2.1	Svarshindrande fel	35
3.2.2	Övriga fel	35
3.3	Principer för versionshantering.....	35
3.3.1	Bakåtkompatibla förändringar	36
3.3.2	Icke-bakåtkompatibla förändringar.....	36
4	Referenser.....	37
5	Appendix A: Exempel kodgenerering av producentanslutning från WSDL.....	37
5.1	Kodgenerering Java.....	37
5.2	Kodgenerering .NET	40

1 Ändringshistorik

Version	Datum	Beskrivning	Ansvarig
1.0	2017-11-07	Gemensamt tekniskt gränssnitt för SSBT oktober 2017.	Jonas Nyfeldt
2.0	2019-05-09	Uppdaterad med information om direkt/indirekt återanvändning och förenklad indirekt återanvändning. Uppdaterat XML-schema informationsmodell företag med juridisk form.	Åsa Ekdahl Erik Tegeback Jonas Nyfeldt
3.0	2019-08-16	Tillägg av datakonsumentens juridiska form i XML-schema metadata för att möjliggöra utlämningsbeslut hos dataproducenter i fallet direkt återanvändning.	Jonas Nyfeldt
4.0	2020-05-15	Lagt till nya informationselement som används i SSBTPR och SSBTGU (UD0050, UD0051 och UD0052)	Malin Berglund Åsa Ekdahl Jonas Nyfeldt

2 Inledning

Vidareförmedling av grundläggande företagsuppgifter består av fyra sammansatta bastjänster: engagemangstjänsten (SSBTEN), rolltjänsten (SSBTRO), tjänsten för grundläggande uppgifter om företag (SSBTGU) och tjänsten för personrelationer i företag (SSBTPR). Dessa tjänster delar XML-scheman och principer för versionshantering och felhantering.

Det här dokumentet beskriver dessa gemensamma delar och används typiskt som referensmaterial vid läsning av referens 1, 2, 3 och 6.

3 Tekniskt gränssnitt

Det här kapitlet beskriver de delar av det tekniska gränssnittet som är gemensamma för förmedlingstjänsterna.

3.1 Gemensamma XML-scheman

De specifika XML-schemana för varje förmedlingstjänst bygger på ett antal gemensamma XML-scheman definierade i detta kapitel.

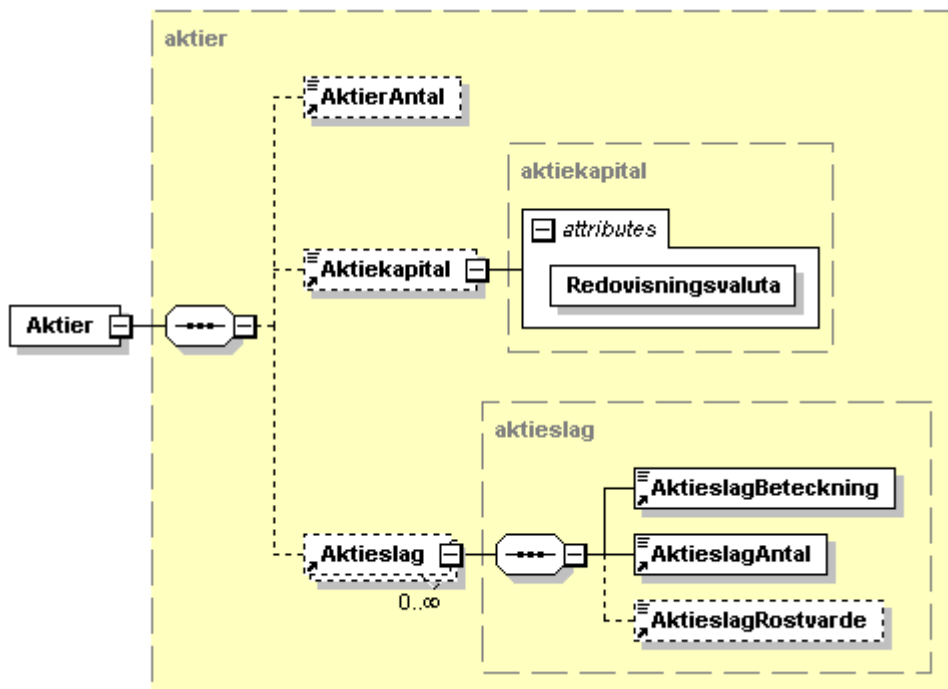
Tabellen nedan beskriver vilka namnrymder som gäller i majorversion 2 av de gemensamma XML-schemana. Tabellen beskriver också vilket prefix som används i det här dokumentet när element ur ett annat XML-schema återanvänds.

XML-schema	Prefix	Namnrymd
Informationsutbytesmodell företag (Foretag.xsd)	iumf	http://schema.bolagsverket.se/ssbt/foretag
Metadata (Metadata.xsd)	md	http://schema.bolagsverket.se/ssbt/metadata
Fel (Fel.xsd)	fel	http://schema.bolagsverket.se/ssbt/fel
Rollintyg (RollIntyg.xsd)	roi	http://schema.bolagsverket.se/ssbt/rollintyg

3.1.1 XML-schema informationsutbytesmodell företag

Det här kapitlet beskriver de element ur informationsutbytesmodellen som bär den nyttoinformation som levereras av producentanslutningarna och förmedlas till konsumentanslutningarna av förmedlingstjänsterna. Element utan prefix hör till namnrymden <http://schema.bolagsverket.se/ssbt/foretag> i det här kapitlet.

3.1.1.1 Aktier



Aktier beskriver företagets aktiekapital och antal aktier, total (AktierAntal) och eventuellt uppdelat per olika aktieslag (AktieslagAntal).

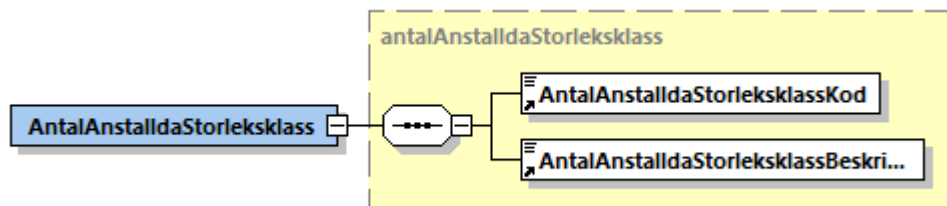
Antal aktier anges som heltal, medan Aktiekapital anges som ett decimaltal. Redovisningsvalutan antar värdet enligt standarden ISO 4217. De valutakoder som är vanligast listas i tabellen nedan:

Värde	Beskrivning
EUR	Euro.
SEK	Svenska kronor.

Exempel för ett företag som har 110000 kronor i aktiekapital fördelat på 1100 aktier, varav 100 är A-aktier med ett röstvärde på 10 och 1000 är B-aktier med ett röstvärde på 1:

```
<Aktier>
  <AktierAntal>1100</AktierAntal>
  <Aktiekapital Redovisningsvaluta="SEK">110000</Aktiekapital>
  <Aktieslag>
    <AktieslagBeteckning>A</AktieslagBeteckning>
    <AktieslagAntal>100</AktieslagAntal>
    <AktieslagRostvarde>10</AktieslagRostvarde>
  </Aktieslag>
  <Aktieslag>
    <AktieslagBeteckning>B</AktieslagBeteckning>
    <AktieslagAntal>1000</AktieslagAntal>
    <AktieslagRostvarde>1</AktieslagRostvarde>
  </Aktieslag>
</Aktier>
```

3.1.1.2 AntalAnstalldaStorleksklass



Antalet anställda i företaget. Anges som storleksklass. Består av en kod för företagets storleksklass och en klartext för att beskriva klassen.

Storleksklasserna för AntalAnstalldaStorleksklass kan anta följande värden:

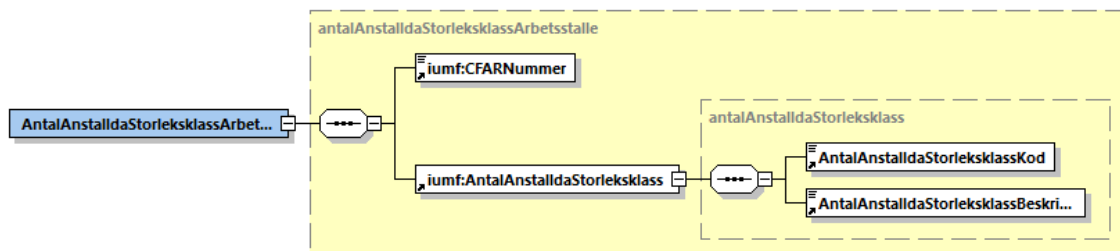
Kod	Beskrivning
1	0 anställda
2	1-4 anställda
3	5-9 anställda
4	10-19 anställda
5	20-49 anställda
6	50-99 anställda
7	100-199 anställda
8	200-499 anställda
9	500-999 anställda
10	1000-1499 anställda
11	1500-1999 anställda
12	2000-2999 anställda
13	3000-3999 anställda
14	4000-4999 anställda
15	5000-9999 anställda
16	10000- anställda

Exempel:

```

<AntalAnstalldaStorleksklass>
  <AntalAnstalldaStorleksklassKod>10</AntalAnstalldaStorleksklassKod>
  <AntalAnstalldaStorleksklassBeskrivning>1000-1499
anställda</AntalAnstalldaStorleksklassBeskrivning>
</AntalAnstalldaStorleksklass>
  
```

3.1.1.3 AntalAnstalldaStorleksklassArbetsstalle



Antalet anställda per arbetsställe, i företaget. Anges som storleksklass. Består av CfarNr för arbetsstället samt en kod för arbetsställets storleksklass och en klartext för att beskriva klassen.

Storleksklasserna för AntalAnstalldaStorleksklassArbetsstalle kan anta följande värden:

Kod	Beskrivning
1	0 anställda
2	1-4 anställda
3	5-9 anställda
4	10-19 anställda
5	20-49 anställda
6	50-99 anställda
7	100-199 anställda
8	200-499 anställda
9	500-999 anställda
10	1000-1499 anställda
11	1500-1999 anställda
12	2000-2999 anställda
13	3000-3999 anställda
14	4000-4999 anställda
15	5000-9999 anställda
16	10000- anställda

Exempel:

```
<AntalAnstalldaStorleksklassArbetsstalle>
  <CFARNummer>00000010</CFARNummer>
  <AntalAnstalldaStorleksklass>
    <AntalAnstalldaStorleksklassKod>3</AntalAnstalldaStorleksklassKod>
    <AntalAnstalldaStorleksklassBeskrivning>5-9
anställda</AntalAnstalldaStorleksklassBeskrivning>
  </AntalAnstalldaStorleksklass>
</AntalAnstalldaStorleksklassArbetsstalle>
```


3.1.1.4 Arbetsgivarregistrerad

Arbetsgivarregistrerad kan anta följande värden:

Värde	Beskrivning
Ja	Företaget är arbetsgivarregistrerat.
Nej	Företaget är inte arbetsgivarregistrerat.

Exempel:

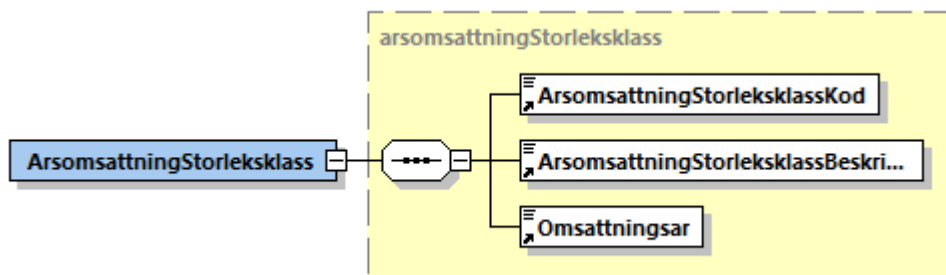
```
<Arbetsgivarregistrerad>Ja</Arbetsgivarregistrerad>
```

3.1.1.5 ArbetstalleBenamning

En benämning för ett arbetsställe. Exempel:

```
<ArbetsstalleBenamning>Bygget</ArbetsstalleBenamning>
```

3.1.1.6 ArsomsattningStorleksklass



Företagets beräknade årsomsättning samt det omsättningsår företagets momsdeklarationer avser, som ligger till grund för årsomsättningen. Årsomsättningen anges som storleksklass.

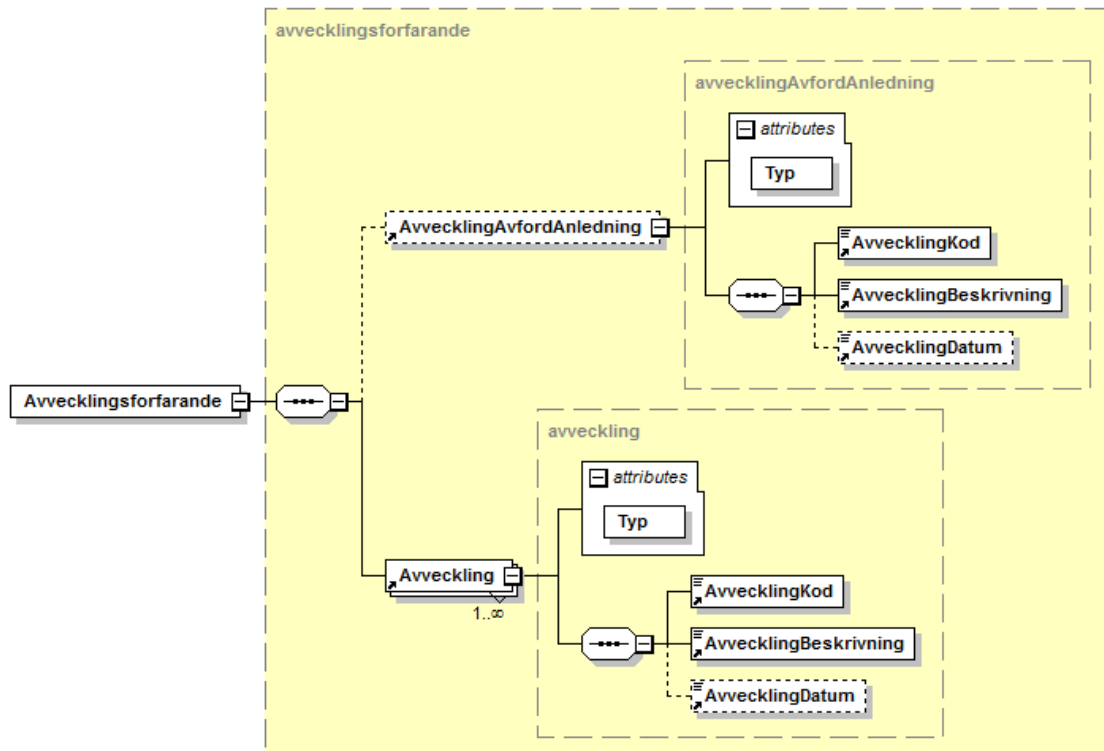
Storleksklasserna för ArsomsattningStorleksklass kan anta följande värden:

Kod	Beskrivning
1	<1 tkr
2	1-49 tkr
3	50-99 tkr
4	100-199 tkr
5	200-299 tkr
6	300-499 tkr
7	500-699 tkr
8	700-999 tkr
9	1000-1499 tkr
10	1500-2999 tkr
11	3000-4999 tkr
12	5000-9999 tkr
13	10000-19999 tkr
14	20000-49999 tkr
15	50000-99999 tkr
16	100000-499999 tkr
17	500000-999999 tkr
18	1000000-4999999 tkr
19	5000000-9999999 tkr
20	>10000000 tkr

Exempel:

```
<ArsomsattningStorleksklass>
  <ArsomsattningStorleksklassKod>16</ArsomsattningStorleksklassKod>
  <ArsomsattningStorleksklassBeskrivning>100000-499999
tkr</ArsomsattningStorleksklassBeskrivning>
  <Omsattningsar>2019</Omsattningsar>
</ArsomsattningStorleksklass>
```

3.1.1.7 Avvecklingsförfarande



Avvecklingsförfarande beskriver status för ett företag under och efter avveckling.

Ett företag som är under avveckling bär information om aktuell avvecklingsstatus i Avveckling-elementen. Efter avförd avveckling sätts också AvvecklingAvfordAnledning för att lyfta fram den händelse som fullbordade avvecklingen. Både Avveckling och AvvecklingAvfordAnledning har ett Typ-attribut som kan anta följande värden:

Värde	Beskrivning
DelningOverlatande	Kod, beskrivning och datum gäller överlåtande vid delning.
DelningOvertagande	Kod, beskrivning och datum gäller övertagande vid delning.
Foretagsrekonstruktion	Kod, beskrivning och datum gäller företagsrekonstruktion.
FusionOverlatande	Kod, beskrivning och datum gäller överlåtande i fusion.
FusionOvertagande	Kod, beskrivning och datum gäller övertagande i fusion.
Konkurs	Kod, beskrivning och datum gäller konkurs.
Likvidation	Kod, beskrivning och datum gäller likvidation.

AvvecklingKod kan anta följande värden:

Värde	Beskrivning
20	Konkurs inledd.
21	Konkurs avslutad.
22	Konkurs avslutad med överskott.
31	Likvidation avslutad.
32	Likvidation beslutad.
33	Likvidation fortsätter.
40	Överlåtande i fusion inledd.
41	Överlåtande i fusion – upplöst genom fusion.
49	Övertagande i fusion pågår.
80	Företagsrekonstruktion inledd.
90	Överlåtande vid delning pågår.
91	Överlåtande vid delning – upplöst genom fusion.
99	Övertagande vid delning pågår.

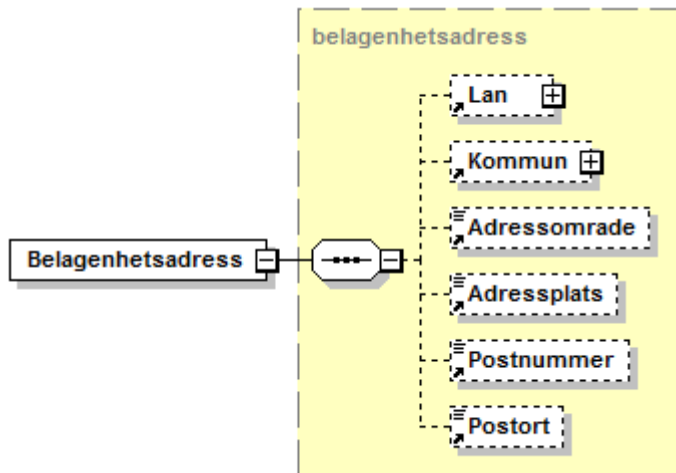
Exempel för ett företag som gått i konkurs med överskott varefter beslut om likvidation fattats (företaget lever fortfarande):

```
<Avvecklingsforfarande>
  <Avveckling Typ="Konkurs">
    <AvvecklingKod>22</AvvecklingKod>
    <AvvecklingBeskrivning>Konkurs avslutad med överskott</AvvecklingBeskrivning>
    <AvvecklingDatum>2014-12-10Z</AvvecklingDatum>
  </Avveckling>
  <Avveckling Typ="Likvidation">
    <AvvecklingKod>32</AvvecklingKod>
    <AvvecklingBeskrivning>Likvidation beslutad</AvvecklingBeskrivning>
    <AvvecklingDatum>2014-12-12Z</AvvecklingDatum>
  </Avveckling>
</Avvecklingsforfarande>
```

Exempel för ett företag som gått i konkurs med överskott varefter beslut om likvidation fattats och avslutats (företaget lever inte längre):

```
<Avvecklingsforfarande>
  <AvvecklingAvfordAnledning Typ="Likvidation">
    <AvvecklingKod>31</AvvecklingKod>
    <AvvecklingBeskrivning>Likvidation avslutad</AvvecklingBeskrivning>
    <AvvecklingDatum>2014-12-19Z</AvvecklingDatum>
  </AvvecklingAvfordAnledning>
  <Avveckling Typ="Konkurs">
    <AvvecklingKod>22</AvvecklingKod>
    <AvvecklingBeskrivning>Konkurs avslutad med överskott</AvvecklingBeskrivning>
    <AvvecklingDatum>2014-12-10Z</AvvecklingDatum>
  </Avveckling>
  <Avveckling Typ="Likvidation">
    <AvvecklingKod>31</AvvecklingKod>
    <AvvecklingBeskrivning>Likvidation avslutad</AvvecklingBeskrivning>
    <AvvecklingDatum>2014-12-19Z</AvvecklingDatum>
  </Avveckling>
</Avvecklingsforfarande>
```

3.1.1.8 Belagenhetsadress



En belägenhetsadress bär information om län, kommun, adressområde, adressplats, postnummer (1-5 siffror) och postort.

Mer detaljerad information om hur län och kommun representeras finns i kapitel 3.1.1.22 respektive kapitel 3.1.1.21.

3.1.1.9 CFARNummer

Ett CFAR-nummer är en unik identitet för ett arbetsställe. CFAR-nummer representeras som 8 siffror. Exempel:

```
<CFARNummer>12345678</CFARNummer>
```

3.1.1.10 EPostAdress

En e-postadress. Exempel:

```
<EPostAdress>exempelbygg@example.com</EPostAdress>
```

3.1.1.11 Firmatecknare

En person kan vara firmatecknare i ett företag. Firmatecknare kan anta följande värden:

Värde	Beskrivning
Ja	Personen är firmatecknare i företaget, på egen hand eller tillsammans med en eller flera andra personer.
Nej	Personen är inte firmatecknare i företaget.

Exempel:

```
<Firmatecknare>Ja</Firmatecknare>
```

3.1.1.12 FirmateckningBeskrivning

En textuell beskrivning av hur företagets firmateckning ser ut. Exempel då firman tecknas av styrelsen:

```
<FirmateckningBeskrivning>Firman tecknas av styrelsen</FirmateckningBeskrivning>
```

Om firman tecknas av flera namngivna personer, kan firmateckningsbeskrivningen till exempel se ut så här:

```
<FirmateckningBeskrivning>Firman tecknas i förening av  
Doe, Jan Jon  
Doe, Jane</FirmateckningBeskrivning>
```

Olika typer av firmateckningar kan också förekomma i kombination. Exempel:

```
<FirmateckningBeskrivning>Firman tecknas av styrelsen  
Firman tecknas i förening av  
Doe, Jan Jon  
Doe, Jane</FirmateckningBeskrivning>
```

3.1.1.13 Fodlesdatum

Födelsedatumet för en fysisk person. Exempel:

```
<Fodlesdatum>1955-01-01</Fodlesdatum>
```

3.1.1.14 ForetagAvregistreratDatum

Datumet då företaget avregistrerades hos ansvarig myndighet. Exempel:

```
<ForetagAvregistreratDatum>2015-01-14Z</ForetagAvregistreratDatum>
```

3.1.1.15 ForetagRegistreratDatum

Datumet då företaget registrerades hos ansvarig myndighet. Exempel:

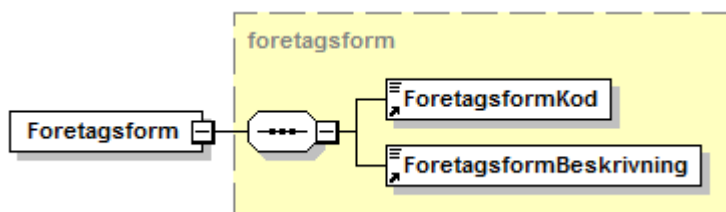
```
<ForetagRegistreratDatum>2012-01-01Z</ForetagRegistreratDatum>
```

3.1.1.16 ForetagNamn

Ett företagsnamn representeras som text i ett ForetagNamn-element. Exempel:

```
<ForetagNamn>Exempelbygg AB</ForetagNamn>
```

3.1.1.17 Företagsform



Ett företags företagsform identifieras av företagsformens kod och beskrivning. Företagsformskoden representeras med en mellan ett och fyra tecken lång textkod.

ForetagsformKod kan anta följande värden:

Värde	Beskrivning
AB	Aktiebolag.
BAB	Bankaktiebolag.
BF	Bostadsförening.
BFL	Utländsk banks filial.
BRF	Bostadsrättsförening.
E	Enskild näringsidkare.
EB	Enkla bolag.
EEIG	Europeisk ekonomisk intressegruppering
EGTS	Europeiska Grupperingar för Territoriellt Samarbete.
EK	Ekonomisk förening.
FAB	Försäkringsaktiebolag.
FL	Filial.
FOF	Försäkringsförening.
HB	Handelsbolag.
I	Ideell förening som bedriver näring.
KB	Kommanditbolag.
KHF	Kooperativ hyresrättsförening.
MB	Medlemsbank.
OFB	Ömsesidigt försäkringsbolag.
S	Stiftelse som bedriver näring.
SB	Sparbank.
SCE	Europakooperativ.
SE	Europabolag.
SF	Sambruksförening.
TSF	Trossamfund.

Exempel:

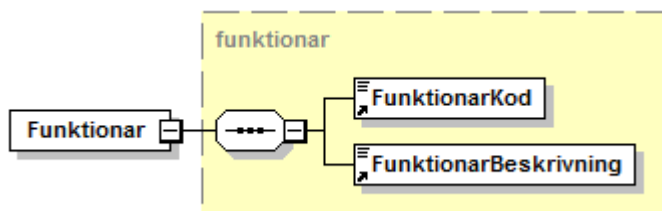
```

<Foretagsform>
  <ForetagsformKod>AB</ForetagsformKod>
  <ForetagsformBeskrivning>aktiebolag</ForetagsformBeskrivning>
</Foretagsform>

<Foretagsform>
  <ForetagsformKod>E</ForetagsformKod>
  <ForetagsformBeskrivning>enskild näringsidkare</ForetagsformBeskrivning>
</Foretagsform>

```

3.1.1.18 Funktionar



En person kan vara en funktionär i ett eller flera företag. På vilket sätt personen är funktionär beskrivs av en funktionärskod. Funktionärskoden representeras med en mellan ett och fem tecken lång textkod.

FunktionärKod kan anta följande värden:

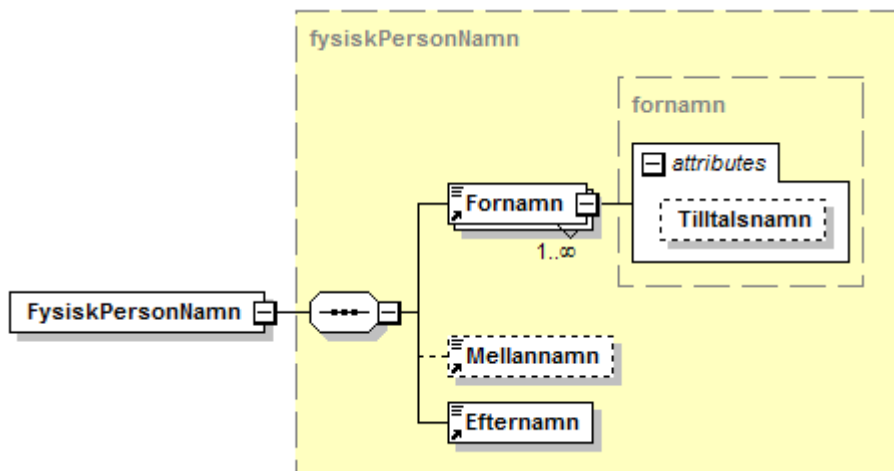
Värde	Beskrivning
AK	Aktuarie.
BO	Bolagsman.
DELG	Delgivningsbar person (särskild delgivningsmottagare).
EFT	Extern firmateknare (ej tillika styrelsemedlem).
EVD	Extern verkställande direktör (ej tillika styrelsemedlem).
EVVD	Extern vice verkställande direktör (ej tillika styrelsemedlem).
FÖ	Föreståndare.
IN	Innehavare.
KD	Kommanditdelägare.
KP	Komplementär.
LE	Styrelseledamot.
LI	Likvidator.
LS	Likvidatorssuppleant.
OF	Ordförande.
PO	Prokurist.
REP	Representant.
REV	Revisor.
REXH	Huvudansvarig revisor.
REVL	Lekmannarevisor.
REVS	Revisorssuppleant.
REVSL	Suppleant för lekmannarevisor.
REVST	Revisorssuppleant (med tillstånd).
REVT	Revisor (med tillstånd).
SU	Suppleant.
SVD	Ställföreträdande verkställande direktör.
VD	Verkställande direktör.
VLE	Verkställande ledamot.
VOF	Vice ordförande.
VVD	Vice verkställande direktör.

Exempel:

```
<Funktionar>
  <FunktionarKod>LE</FunktionarKod>
  <FunktionarBeskrivning>styrelseledamot</FunktionarBeskrivning>
</Funktionar>

<Funktionar>
  <FunktionarKod>IN</FunktionarKod>
  <FunktionarBeskrivning>innehavare</FunktionarBeskrivning>
</Funktionar>
```


3.1.1.19 FysiskPersonNamn



Ett namn på en fysisk person bär information om personens förnamn, efternamn och eventuellt mellannamn.

Tilltalsnamn kan anta följande värden:

Värde	Beskrivning
Ja	Förnamnet är personens tilltalsnamn.
Nej	Förnamnet är inte personens tilltalsnamn. Samma betydelse uppnås också genom att utelämna Tilltalsnamn-attributet (rekommenderas).

För personer med bara ett förnamn kan förnamnet levereras med eller utan tilltalsnamnsmaking, dataproducenten bestämmer. Ordningen på Fornamn speglar, i möjligaste mån, personens verkliga namnordning. Högst ett förnamn ska markeras med Tilltalsnamn = Ja.

XML-representationen för några exempelnamn följer. Exempelnamnens tilltalsnamn skrivs med fetstil. Exempelnamnens mellannamn skrivs med kursiv stil.

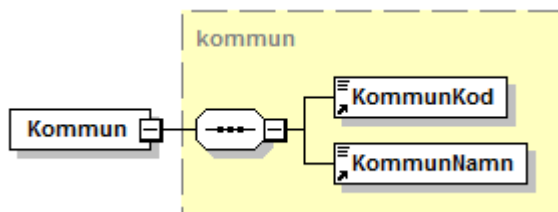
Exempelnamn	Beskrivning
Anna Bengtsson	<FysiskPersonNamn> <Fornamn>Anna</Fornamn> <Efternamn>Bengtsson</Efternamn> </FysiskPersonNamn>
Anna Beata Cederlund	<FysiskPersonNamn> <Fornamn>Anna</Fornamn> <Fornamn Tilltalsnamn="Ja">Beata</Fornamn> <Efternamn>Cederlund</Efternamn> </FysiskPersonNamn>
Anna-Beata Cecilia Davidsson	<FysiskPersonNamn> <Fornamn Tilltalsnamn="Ja">Anna-Beata</Fornamn> <Fornamn>Cecilia</Fornamn> <Efternamn>Davidsson</Efternamn> </FysiskPersonNamn>
Anna Beata Cecilia Davidsson	<FysiskPersonNamn> <Fornamn Tilltalsnamn="Ja">Anna Cecilia</Fornamn> <Fornamn>Beata</Fornamn> <Efternamn>Davidsson</Efternamn> </FysiskPersonNamn>
Anna Beata-Cecilia <i>Davidsson</i> Eriksson	<FysiskPersonNamn> <Fornamn Tilltalsnamn="Ja">Anna</Fornamn> <Fornamn>Beata-Cecilia</Fornamn> <Mellannamn>Davidsson</Mellannamn> <Efternamn>Eriksson</Efternamn> </FysiskPersonNamn>

3.1.1.20 Juridisk form

Företagets juridiska form. Koden för juridisk form är en tvåställig numerisk kod. Exempel:

```
<JuridiskForm>
  <JuridiskFormKod>49</JuridiskFormKod>
  <JuridiskFormBeskrivning>Övriga aktiebolag</JuridiskFormBeskrivning>
</JuridiskForm>
```

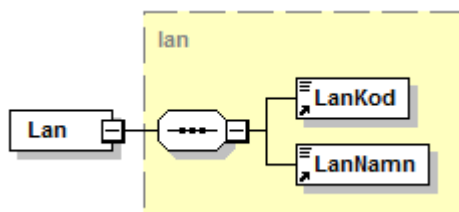
3.1.1.21 Kommun



En kommun identifieras av kommunens kod och namn. Kommunskoden är en fyrställig kod som publiceras av SCB. Exempel:

```
<Kommun>
  <KommunKod>1880</KommunKod>
  <KommunNamn>Örebro</KommunNamn>
</Kommun>
```

3.1.1.22 Lan



Ett län identifieras av länets kod och namn. Länskoderna är en tvåställig kod som publiceras av SCB. Exempel:

```
<Lan>
  <LanKod>18</LanKod>
  <LanNamn>Örebro län</LanNamn>
</Lan>
```

3.1.1.23 Momsregistrerad

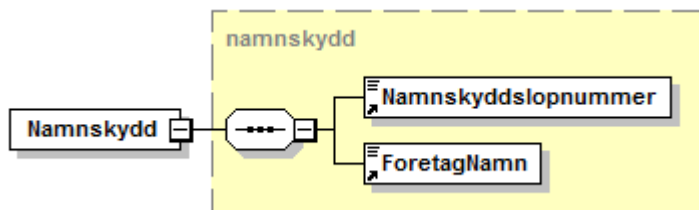
Momsregistrerad kan anta följande värden:

Värde	Beskrivning
Ja	Företaget är momsregistrerat.
Nej	Företaget är inte momsregistrerat.

Exempel:

```
<Momsregistrerad>Ja</Momsregistrerad>
```

3.1.1.24 Namnskydd

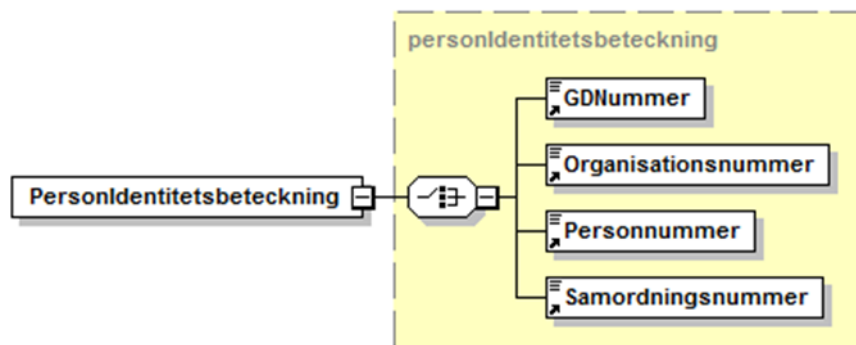


I de fall en enskild näringsidkare har ett namnskyddat företagsnamn associeras detta företagsnamn med ett namnskyddslopnummer mellan 1 och 999. Exempel:

```

<Namnskydd>
  <Namnskyddslopnummer>1</Namnskyddslopnummer>
  <ForetagNamn>Jons blommor och blad</ForetagNamn>
</Namnskydd>
  
```

3.1.1.25 Personidentitetsbeteckning



En person identifieras av organisationsnummer (juridisk person), personnummer (fysisk person), samordningsnummer eller GD-nummer. Organisationsnummer representeras som 10 siffror, personnummer representeras som 12 siffror (ååååmmddnnnn), samordningsnummer representeras som 12 siffror (ååååmmDDnnnn, där DD är personens födelsedag + 60) och GD-nummer representeras som 12 siffror (16nnnnnnnnnn).

Exempel:

```

<PersonIdentitetsbeteckning>
  <Organisationsnummer>5561234567</Organisationsnummer>
</PersonIdentitetsbeteckning>
  
```

```

<PersonIdentitetsbeteckning>
  <Personnummer>198001011234</Personnummer>
</PersonIdentitetsbeteckning>
  
```

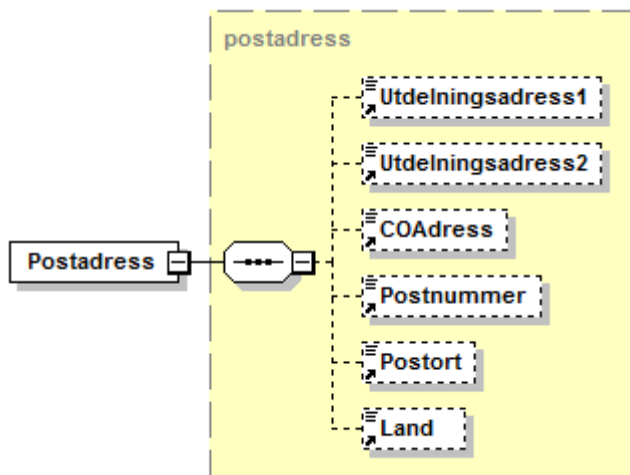
```

<PersonIdentitetsbeteckning>
  <Samordningsnummer>198001611234</Samordningsnummer>
</PersonIdentitetsbeteckning>
  
```

```

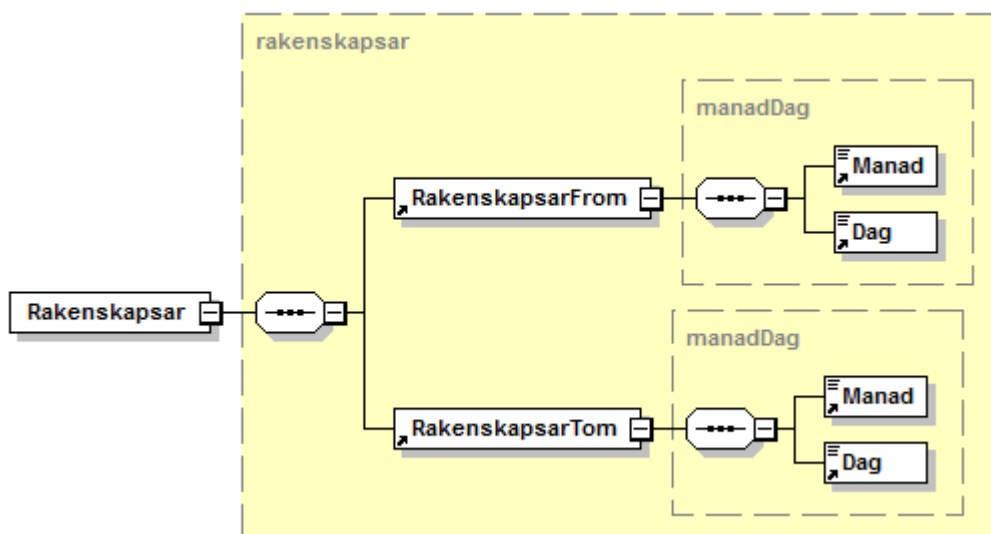
<PersonIdentitetsbeteckning>
  <GDNummer>163021234567</GDNummer>
</PersonIdentitetsbeteckning>
  
```

3.1.1.26 Postadress



En postadress bär information om utdelningsadresser, C/O-adress, postnummer (1-5 siffror), postort och land.

3.1.1.27 Rakenskapsar



Räkenskapsåret definierat som en period fr.o.m. en månad och dag t.o.m. en annan, senare månad och dag. Exempel:

```
<Rakenskapsar>
  <RakenskapsarFrom>
    <Manad>3</Manad>
    <Dag>1</Dag>
  </RakenskapsarFrom>
  <RakenskapsarTom>
    <Manad>2</Manad>
    <Dag>28</Dag>
  </RakenskapsarTom>
</Rakenskapsar>
```

Notera att information om ett räkenskapsår levereras utan hänsyn till skottår.

3.1.1.28 Skatteform

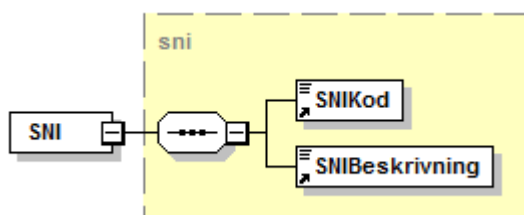
Skatteform kan anta följande värden:

Värde	Beskrivning
Ja	Företaget är godkänt för F- eller FA-skatt.
Nej	Företaget är inte godkänt för F- eller FA-skatt.

Exempel:

```
<Skatteform>Ja</Skatteform>
```

3.1.1.29 SNI



Information om svensk näringsindelning (SNI) representeras av en kod och en beskrivning. SNI-koden representeras med fem siffror.

SNI-koderna definieras i ett Excel-dokument i fliken ”Detaljgrupp (Femsiffror)”, se referens 4. Motsvarande information finns också i ett PDF-dokument, se referens 5.

Exempel:

```
<SNI>
  <SNIKod>41200</SNIKod>
  <SNIBeskrivning>Byggnad av hus</SNIBeskrivning>
</SNI>
```

```
<SNI>
  <SNIKod>43120</SNIKod>
  <SNIBeskrivning>Mark- och grundarbeten</SNIBeskrivning>
</SNI>
```

```
<SNI>
  <SNIKod>68201</SNIKod>
  <SNIBeskrivning>Uthyrning och förvaltning av egna eller arrenderade
  bostäder</SNIBeskrivning>
</SNI>
```

3.1.1.30 Telefonnummer

Ett telefonnummer. Exempel:

```
<Telefonnummer>0701234567</Telefonnummer>
```

3.1.1.31 Verksamhetsbeskrivning

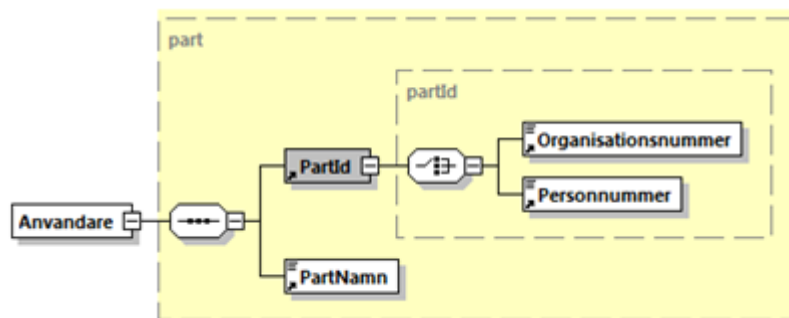
Företagets verksamhetsbeskrivning. Exempel:

```
<Verksamhetsbeskrivning>Bolaget bedriver verksamhet inom området byggnation och därmed
förenlig verksamhet</Verksamhetsbeskrivning>
```

3.1.2 XML-schema metadata

Det här kapitlet beskriver de element som bär information om metadata i begäran- och svarsmeddelanden mellan konsumentanslutningar och förmedlingstjänster samt mellan förmedlingstjänster och producentanslutningar. Element utan prefix hör till namnrymden <http://schema.bolagsverket.se/ssbt/metadata> i det här kapitlet.

3.1.2.1 Användare



Användare vid indirekt återanvändning

En datakonsument som vill använda informationen i en e-tjänst å en autentiserad och auktoriserad användares vägnar sätter Användare till inloggad person för att identifiera användaren (PartId och PartNamn). Exempel:

```
<Användare>
  <PartId>
    <Personnummer>198001011234</Personnummer>
  </PartId>
  <PartNamn>Jon Doe</PartNamn>
</Användare>
```

Användare vid direkt återanvändning

En datakonsument som vill använda informationen i ett verksamhetssystem sätter datakonsumenten som Användare (PartId och PartNamn). Exempel:

```
<Användare>
  <PartId>
    <Organisationsnummer>2021001234</Organisationsnummer>
  </PartId>
  <PartNamn>Datakonsumentverket</PartNamn>
</Användare>
```

3.1.2.2 Användningsområde

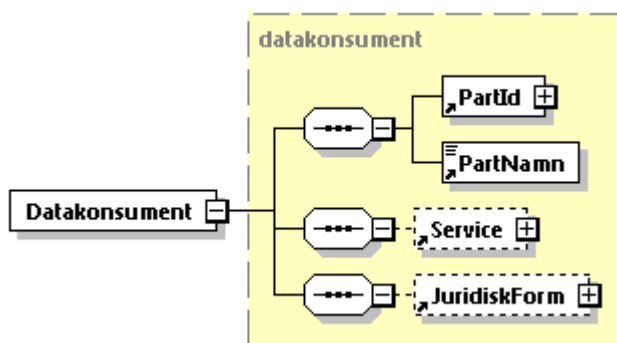
En sändare som agerar i rollen datakonsument sätter användningsområde till direkt eller indirekt återanvändning enligt följande enumeration:

Värde	Beskrivning
DirektAteranvandning	Direkt återanvändning – om informationen ska användas i ett verksamhetssystem.
IndirektAteranvandning	Indirekt återanvändning – om informationen ska användas i en e-tjänst.

Exempel:

```
<Anvandsomrade>DirektAteranvandning</Anvandsomrade>
```

3.1.2.3 Datakonsument

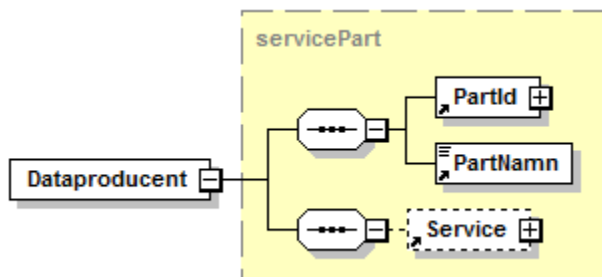


En sändare som agerar i rollen datakonsument sätter Datakonsument för att identifiera sig själv (PartId och PartNamn) och anropande tjänst (Service). Exempel:

```
<Datakonsument>
  <PartId>
    <Organisationsnummer>2021005489</Organisationsnummer>
  </PartId>
  <PartNamn>Bolagsverket</PartNamn>
  <Service>
    <ServiceNamn>Verksamt</ServiceNamn>
  </Service>
</Datakonsument>
```

En datakonsument ska aldrig sätta JuridiskForm eftersom detta, i förekommande fall, sätts av förmedlaren enligt datakonsumentens anslutningsavtal.

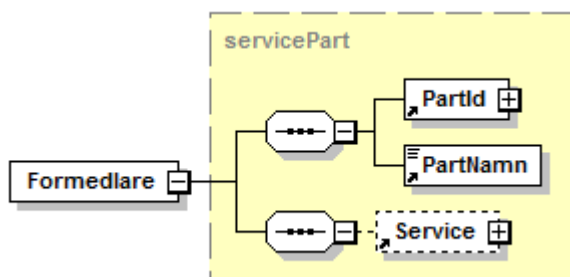
3.1.2.4 Dataproducent



En sändare som agerar i rollen dataproducent sätter Dataproducent för att identifierar sig själv (PartId och PartNamn) och producerande tjänst (Service). Exempel:

```
<Dataproducent>
  <PartId>
    <Organisationsnummer>2021005489</Organisationsnummer>
  </PartId>
  <PartNamn>Bolagsverket</PartNamn>
  <Service>
    <ServiceNamn>BATGU</ServiceNamn>
  </Service>
</Dataproducent>
```

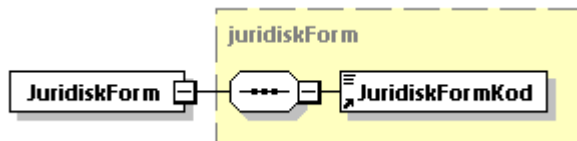
3.1.2.5 Förmedlare



En sändare som agerar i rollen förmedlare sätter Förmedlare för att identifierar sig själv (PartId och PartNamn) och förmedlande tjänst (Service). Exempel:

```
<Förmedlare>
  <PartId>
    <Organisationsnummer>2021005489</Organisationsnummer>
  </PartId>
  <PartNamn>Bolagsverket</PartNamn>
  <Service>
    <ServiceNamn>SSBTGU</ServiceNamn>
  </Service>
</Förmedlare>
```

3.1.2.6 JuridiskForm



Juridisk form identifieras av en tvåställig numerisk kod. Exempel:

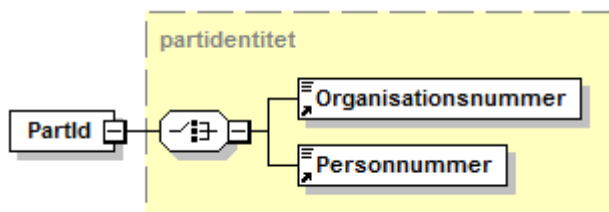
```
<JuridiskForm>
  <JuridiskFormKod>81</JuridiskFormKod>
</JuridiskForm>
```

3.1.2.7 Meddelandeld

UUID för ett meddelande (begäran eller svar). Sändare skapar en ny unik identitet för varje meddelande, vilket ger möjlighet att logga MeddelandeId hos både sändare och mottagare för spårbarhet. Exempel:

```
<MeddelandeId>e4952ff4-3a83-45df-b408-7e437e838fba</MeddelandeId>
```

3.1.2.8 PartId

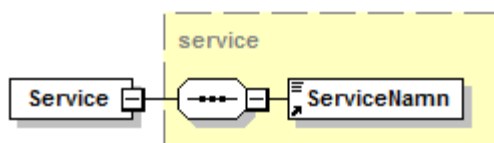


En part identifieras antingen av ett organisationsnummer eller av ett personnummer. Personnummer representeras som 12 siffror (ååååmmddnnnn). Organisationsnummer representeras som 10 siffror. Exempel:

```
<PartId>
  <Personnummer>198001011234</Personnummer>
</PartId>
```

```
<PartId>
  <Organisationsnummer>5561234567</Organisationsnummer>
</PartId>
```

3.1.2.9 Service



En Service identifieras via ett ServiceNamn. Exempel:

```
<Service>
  <ServiceNamn>Verksamt</ServiceNamn>
</Service>
```

3.1.2.10 Tidstempel

En tidstempel med millisekundprecision. Exempel:

```
<Tidstempel>2014-12-01T09:30:48.123Z</Tidstempel>
```

3.1.2.11 TransaktionId

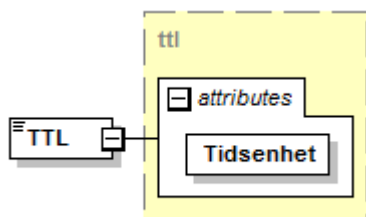
UUID för transaktion. Mottagare svarar alltid med samma TransaktionId som sändaren satt i sin begäran. Används av mottagare för att säkerställa att svaret gäller för den begäran som skickats.

Exempel:

```
<TransaktionId>3cf72268-246a-4f68-a296-40b202e4240e</TransaktionId>
```

SSBTGU associerar i sina loggar datakonsumentens TransaktionId med det TransaktionId som skickas till dataproducenterna i syfte att stödja spårbarhet datakonsument-SSBTGU-dataproducent.

3.1.2.12 TTL



En sändare kan sätta en Time-To-Live på en begäran för att informera en mottagare om hur länge sändaren är beredd att vänta på ett svar. Mottagare uppmuntras att, inom rimliga gränser, hedra en sändares begäran om TTL. Mottagare uppmuntras också att sätta ett tak (och eventuellt också ett golv) för att skydda sig själv mot orimliga önskade svarstider. Sändare som inte sätter en TTL överlåter till mottagaren att bestämma TTL för begäran.

Attributet Tidsenhet berättar i vilken tidsenhet TTL-värdet är angivet. Tillsvidare stöds bara ett värde:

Värde	Beskrivning
Millisekunder	TTL-värdet är angivet i millisekunder.

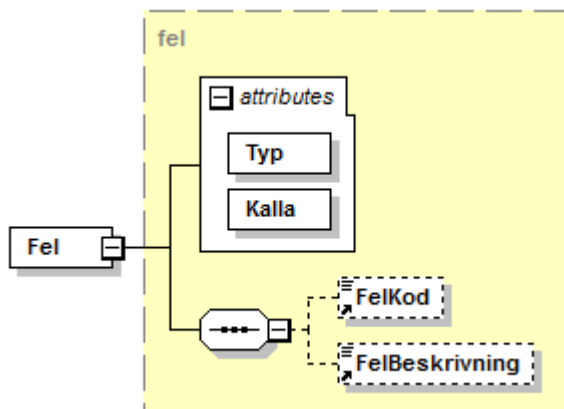
Exempel:

```
<TTL Tidsenhet="Millisekunder">3000</TTL>
```

3.1.3 XML-schema fel

Det här kapitlet beskriver de element som bär information om fel mellan producentanslutningar och förmedlingstjänster samt mellan förmedlingstjänster och konsumentanslutningar. Element utan prefix hör till namnrymden <http://schema.bolagsverket.se/ssbt/fel> i det här kapitlet.

3.1.3.1 Fel och feltyp



Ett fel beskrivs primärt av en feltyp (Typ). En feltyp sätts för alla fel och presenteras av datakonsumentens e-tjänster om behovet finns att informera användaren om felen.

Attributet Typ beskriver felets feltyp:

Värde	Beskrivning
EjBehorig	Användaren är inte behörig att se efterfrågade uppgifter för begärt företag.
EjKomplettSvar	Dataproducenten har inte fått ett komplett svar från underliggande verksamhetssystem/register och kan därför inte leverera en tillförlitlig uppgift.
ForetagFinnsEj	Begärt företag finns inte registrerat i sökbar form hos aktuell dataproducent. Observera att ett företag ändå kan existera och eventuellt också finnas registrerat i sökbar form hos en annan dataproducent trots detta fel.
OgiltigBegaran	Begäran kan inte hanteras av mottagaren. Exempel: <ul style="list-style-type: none"> • XML-valideringsfel. • Datakonsumentens organisationsnummer finns inte i förmedlingstjänstens register över anslutna datakonsumenter. • Dataproducenten hanterar inte frågor som rör direkt återanvändning.
OgiltigtSvar	Svar kan inte hanteras av mottagaren, till exempel på grund av XML-valideringsfel.
OtillgangligUppgiftskälla	Tjänsten kan inte nå uppgiftskällan. I förmedlingstjänsternas fall avser uppgiftskälla en producentanslutning och i producentanslutningens fall avser uppgiftskälla ett underliggande verksamhetssystem/register.
Timeout	Svar levererades inte inom begärd svarstid (TTL).
TvetydigBegaran	Frågan är för öppet ställd för att ge ett entydigt svar.
UppgiftFinnsEj	Dataproducenten kan inte leverera efterfrågad uppgift för begärt företag trots att den alltid borde finnas. Om det kan anses normalt att uppgiften saknas för begärt företag, ska dataproducenten inte signalera ett Fel utan istället utelämna uppgiftselementet.
Odefinierad	Feltypen är odefinierad, d.v.s. felet är av en typ som inte kan beskrivas av någon av de andra, mer specifika feltyperna.

Ett fel bär också information om felkälla (Kalla), d.v.s. vilken tjänst som har upptäckt och skapat Fel-elementet i svarsmeddelandet.

Attributet Kalla beskriver felets felkälla:

Värde	Beskrivning
Dataproducent	Dataproducenten upptäckte felet.
Förmedlare	Förmedlaren upptäckte felet.

Till sist *kan* också felet bära information om en FelKod och en FelBeskrivning. Notera dock att datakonsumenters e-tjänster *aldrig* ska presentera sådan information till användare. En mer detaljerad beskrivning av användningsområdena för FelKod och FelBeskrivning följer i kapitel 3.1.3.2.

Fel som levereras av dataproducenter vidarebefordras till datakonsumenter av förmedlingstjänsterna utan några förändringar. Rekommendationer om hur datakonsumenter bör presentera olika typer av fel finns i kapitel 3.2.

3.1.3.2 FelKod och FelBeskrivning

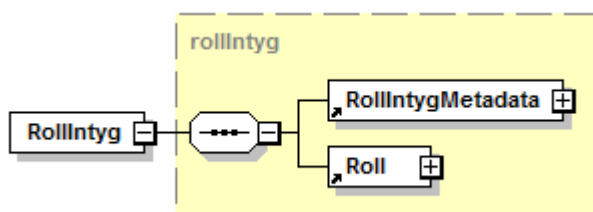
Varje tjänst har möjlighet att definiera sina egna, interna felkoder och/eller felbeskrivningar för att på ett finkornigt och tydligt sätt kunna beskriva olika fel i syfte att möjliggöra spårbarhet och ge bättre möjligheter till felsökning.

Vare sig felkoder eller felbeskrivningar är obligatoriska, men rekommendationen är att åtminstone skicka med felbeskrivningar för alla fel.

3.1.4 XML-schema rollintyg

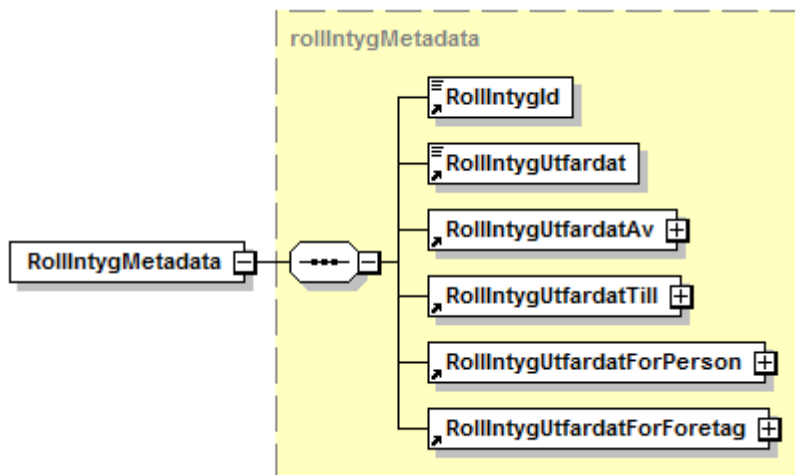
Det här kapitlet beskriver de element som bär information om en användares roll i ett företag i form av ett rollintyg. Element utan prefix hör till namnrymden <http://schema.bolagsverket.se/ssbt/rollintyg> i det här kapitlet.

3.1.4.1 RollIntyg



Ett rollintyg består av en metadatadel (RollIntygMetadata) och en uppgiftsdel (Roll).

3.1.4.2 RollIntygMetadata



Metadata om rollintyget: en identitet, en tidstämpel som visar när intyget utfärdades, vilken dataproducent som utfärdat intyget, vilken datakonsument som intyget utfärdats till, samt för vilken person och företag som intyget gäller.

3.1.4.3 RollIntygId

UUID för ett rollintyg. Dataproducenter till SSBTRO skapar en ny unik identitet för varje utfärdat rollintyg, vilket ger möjlighet att spåra rollintygets färd från SSBTRO-dataproducent till datakonsument och vidare från datakonsument till SSBTGU-dataproducenter.

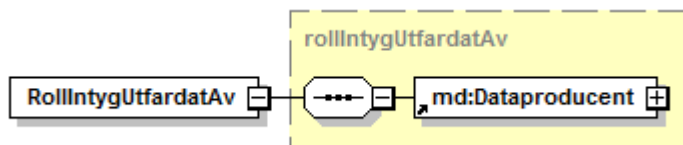
```
<RollIntygId>f2228a1b-de8c-4c2c-8c33-17a3b104fa26</RollIntygId>
```

3.1.4.4 RollIntygUtfardat

En tidstämpel med millisekundprecision som beskriver när rollintyget utfärdats. Exempel:

```
<RollIntygUtfardat>2016-04-27T12:30:02.520z</RollIntygUtfardat>
```

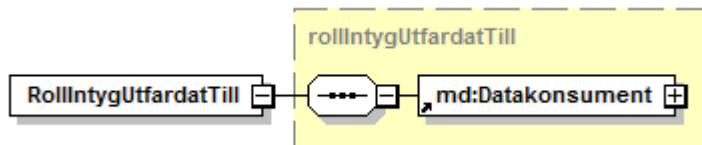
3.1.4.5 RollIntygUtfardatAv



Den dataproducent som utfärdat rollintyget. Exempel:

```
<RollIntygUtfardatAv>
  <md:Dataproducent>
    <md:PartId>
      <md:Organisationsnummer>2021005489</md:Organisationsnummer>
    </md:PartId>
    <md:PartNamn>Bolagsverket</md:PartNamn>
  </md:Dataproducent>
</RollIntygUtfardatAv>
```

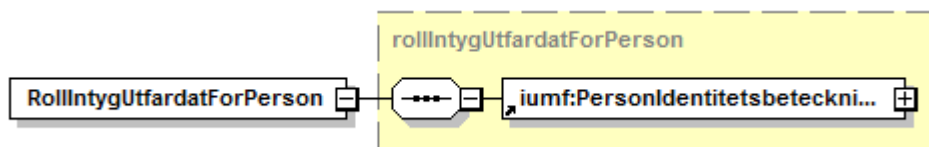
3.1.4.6 RollIntygUtfardatTill



Den datakonsument som rollintyget är utfärdat till. Exempel:

```
<RollIntygUtfardatTill>
  <md:Datakonsument>
    <md:PartId>
      <md:Organisationsnummer>2021001234</md:Organisationsnummer>
    </md:PartId>
    <md:PartNamn>Datakonsumentverket</md:PartNamn>
    <md:Service>
      <md:ServiceNamn>E-tjänst1</md:ServiceNamn>
    </md:Service>
  </md:Datakonsument>
</RollIntygUtfardatTill>
```

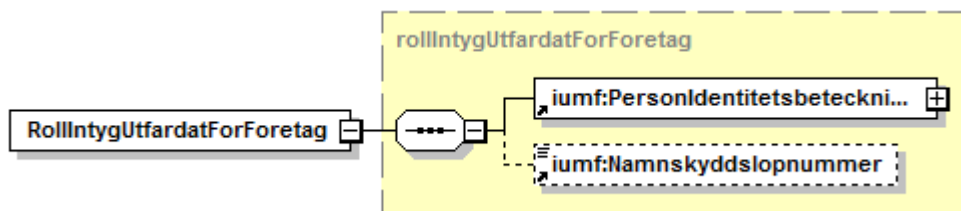
3.1.4.7 RollIntygUtfardatForPerson



Den person som rollintyget är utfärdat för (i kombination med RollIntygUtfardatForForetag, se kapitel 3.1.4.8). Exempel:

```
<RollIntygUtfardatForPerson>
  <iumf:Personidentitetsbeteckning>
    <iumf:Personnummer>198001011234</iumf:Personnummer>
  </iumf:Personidentitetsbeteckning>
</RollIntygUtfardatForPerson>
```

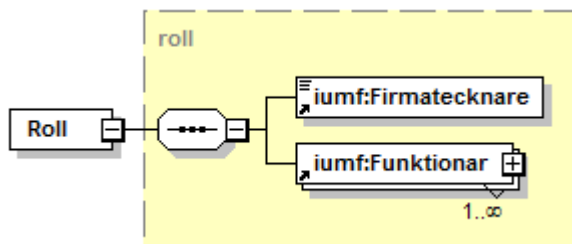
3.1.4.8 RollIntygUtfardatForForetag



Det företag som rollintyget är utfärdat för (i kombination med RollIntygUtfardatForPerson, se kapitel 3.1.4.7). Exempel

```
<RollIntygUtfardatForForetag>
  <iumf:Personidentitetsbeteckning>
    <iumf:Organisationsnummer>5561234567</iumf:Organisationsnummer>
  </iumf:Personidentitetsbeteckning>
</RollIntygUtfardatForForetag>
```


3.1.4.9 Roll



Den roll en person har i ett företag. Exempel:

```

<Roll>
  <iumf:Firmatecknare>Ja</iumf:Firmatecknare>
  <iumf:Funktionar>
    <iumf:FunktionarKod>LE</iumf:FunktionarKod>
    <iumf:FunktionarBeskrivning>styrelseledamot</iumf:FunktionarBeskrivning>
  </iumf:Funktionar>
  <iumf:Funktionar>
    <iumf:FunktionarKod>VD</iumf:FunktionarKod>
    <iumf:FunktionarBeskrivning>verkställande direktör</iumf:FunktionarBeskrivning>
  </iumf:Funktionar>
</Roll>

```

3.2 Felhantering

Fel definieras utifrån dataproducenternas perspektiv. Om en dataproducent inte kan leverera efterfrågade uppgifter kommer detta kommuniceras som ett fel, oavsett bakomliggande orsak till att uppgiften inte kunde levereras.

Från datakonsumentens perspektiv bör dock inte alla typer av fel hanteras på samma sätt gentemot användaren. Följande tabell visar en kategorisering av de feltyper som definierats i kapitel 3.1.3.1 och hur datakonsumentens e-tjänst bör presentera fel i respektive kategori.

Felkategori	Beskrivning felkategori	Feltyper i felkategori
Applikationsfel	<p>Någon tjänst (konsumentanslutning, förmedlingstjänst eller producentanslutning) har skickat ett meddelande som inte kan hanteras av mottagaren.</p> <p>Datakonsumentens e-tjänst bör informera användaren om felet, dock är det osannolikt att omförsök lyckas utan att den felande tjänsten uppgraderats.</p>	OgiltigBegaran
		OgiltigtSvar
		Odefinierad
Tekniska fel	<p>Någon tjänst (förmedlingstjänst eller producentanslutning) har tillfälligt drabbats av tekniska problem, såsom nätverksproblem eller otillgängliga tjänster.</p> <p>Datakonsumentens e-tjänst bör informera användaren om felet och uppmuntra till omförsök.</p>	OtillgängligUppgiftskalla
		Timeout
Användningsfel	<p>Datakonsumentens e-tjänst har frågat efter uppgifter som inte kan levereras.</p> <p>Datakonsumentens e-tjänst bör informera användaren om att uppgifterna inte kunde levereras, <i>utan att</i> visuellt signalera det som ett <i>fel</i>.</p> <p>Utformning av e-tjänstens informationsmeddelanden bör göras i samråd med den dataproducent som levererar de berörda uppgifterna.</p>	EjBehorig
		EjKomplettSvar
		ForetagFinnsEj
		TvetydigBegaran
		UppgiftFinnsEj

Hur ett fel kommuniceras beror på om felet är svarshindrande eller ej. Hanteringen av svarshindrande fel beskrivs i kapitel 3.2.1 medan ej svarshindrande fel beskrivs i kapitel 3.2.2.

3.2.1 Svarshindrande fel

Ett svarshindrande fel definieras som ett fel där en mottagare (förmedlingstjänst eller producentanslutning) inte kan leverera alla efterfrågade uppgifter.

Vid svarshindrande fel ska mottagaren svara med ett SOAP-fault med ett av respektive förmedlingstjänst definierat Servicefel.

3.2.2 Övriga fel

Ett övrigt fel definieras som ett fel som inte gäller alla efterfrågade uppgifter, till exempel att någon uppgiftskälla är otillgänglig eller att en viss uppgift inte kan utlämnas på grund av en dataproducers regelverk för utlämning.

Vid övriga fel ska mottagaren svara som vanligt, med skillnaden att de uppgifter som inte kan levereras kommer innehålla ett Fel enligt kapitel 3.1.3 istället för de efterfrågade uppgifterna.

3.3 Principer för versionshantering

En grundläggande princip för versionshantering av gränssnittet är att en ny version av gränssnittet först implementeras av förmedlingstjänsten. Konsumentanslutningar och producentanslutningar kan sedan uppgraderas samtidigt eller senare.

Förändringar av gränssnittet klassas i bakåtkompatibla respektive icke-bakåtkompatibla förändringar sett från anslutningarnas perspektiv. Gränssnittets version dokumenteras med en treställigt versionsnummer (majorversion.minorversion.patchversion) enligt följande exempel:

- Exempel WSDL-fil:
`<wsdl:documentation>SSBTGU version 1.1.0</wsdl:documentation>`
- Exempel XML-schema:
`<xs:schema ... version="1.1.0">...</xs:schema>`

En ny version av WSDL/XML-scheman där alla förändringar är bakåtkompatibla innebär att XML-schemats patchversion eller minorversion stegas upp. Patchversionen stegas upp (till exempel från version 1.0.0 till version 1.0.1) vid förändringar som inte introducerar någon ny funktionalitet. Att stega patchversion kommer således typiskt vara ovanligt, men kan ske till exempel vid tillägg av kommentarer (`<!-- ... -->` eller `<xs:annotation>...</xs:annotation>`) i WSDL/XML-scheman. Minorversionen stegas upp (till exempel från version 1.0.1 till version 1.1.0) vid förändringar som introducerar ny funktionalitet utan att bryta bakåtkompatibilitet. Hur en bakåtkompatibel förändring klassificeras och hanteras beskrivs mer detaljerat i kapitel 3.3.1.

Är någon förändring icke-bakåtkompatibel innebär det att WSDL/XML-schemans majorversion stegas upp (till exempel från version 1.1.0 till version 2.0.0). Hur en icke-bakåtkompatibel förändring klassificeras och hanteras beskrivs mer detaljerat i kapitel 3.3.2.

Notera också att förmedlingstjänsterna tillämpar strikt schemavalidering både vid mottagning av begäran från datakonsumenter och vid mottagning av svar från

dataproducenter. Detta innebär att datakonsumenter och dataproducenter måste följa WSDL/XML-schemana till punkt och pricka, vilket också omfattar att inte skicka okända element, attribut eller enumerationsvärden till förmedlingstjänsterna.

3.3.1 Bakåtkompatibla förändringar

För att en ny version av gränssnittet ska klassificeras som bakåtkompatibel krävs först och främst att alla befintliga element och attribut har bibehållen semantik, det vill säga att alla värden som kommuniceras i ett element eller attribut har samma betydelse i den nya versionen som i den gamla.

För att en förändring ska klassificeras som bakåtkompatibel krävs också att alla datakonsumenter och dataproducenter kan fortsätta att skicka och ta emot alla meddelanden *via befintliga URLer* även efter att förmedlingstjänsten uppgraderas till en ny version av gränssnittet.

3.3.2 Icke-bakåtkompatibla förändringar

Om någon förändring av WSDL/XML-schema inte kan klassificeras som bakåtkompatibel blir den nya versionen av gränssnittet klassificerad som icke-bakåtkompatibel.

Detta innebär att berörda förmedlingstjänster kommer börja stödja den nya majorversionen *via nya URLer* samtidigt som tidigare majorversioner fortsätter stödjas *via befintliga URLer*. Detta ger datakonsumenter och dataproducenter möjligheten att i egen takt uppgradera till en ny majorversion, till priset av en mer komplicerad hantering i berörda förmedlingstjänster. När alla datakonsumenter och dataproducenter lämnat en tidigare majorversion kan stödet för den versionen avvecklas i berörda förmedlingstjänster.

Att hålla systemlösningen så enkel som möjligt underlättar uppfyllandet av icke-funktionella krav såsom stabilitet och prestanda. Därför är en princip för versionshantering av gränssnittet att sträva efter att hålla bakåtkompatibilitet när så är praktiskt möjligt. Denna strävan kan uttrycka sig på flera sätt, inklusive att välja att koordinera en driftsättning av en ny version av en förmedlingstjänst med berörda dataproducenter och/eller datakonsumenter för att då kunna klassificera en förändring som bakåtkompatibel istället för icke-bakåtkompatibel.

Förändringsloggarna i referens 1, 2 och 3 beskriver om - och i så fall hur – införandet av varje enskild förändring löses med bibehållen bakåtkompatibilitet.

4 Referenser

Referenser:

1. Teknisk guide SSBTGU.
2. Teknisk guide SSBTEN.
3. Teknisk guide SSBTRO.
4. SNI 2007:
http://www.scb.se/Grupp/Hitta_statistik/Forsta_Statistik/Klassifikationer/_Dokument/SNI/SNI2007.xlsx
5. Kortversion SNI 2007:
http://www.scb.se/Grupp/Hitta_statistik/Forsta_Statistik/Klassifikationer/_Dokument/SNI/SNI2007-kortversion.pdf
6. Teknisk guide SSBTPR.

5 Appendix A: Exempel kodgenerering av producentanslutning från WSDL

Eftersom förmedlingstjänsterna applicerar strikt XML-schemavalidering av svar från producentanslutningarna är det viktigt att all data levereras enligt respektive förmedlingstjänsts WSDL och XML-scheman.

Det rekommenderade sättet att leverera korrekta data är att generera kod ”contract-first” från WSDL samt använda XML-schemavalidering. Hur kod genereras från en WSDL-fil och hur producentanslutningen sedan exponerar denna WSDL-fil mot SSBTGU är plattformsb beroende. Stödet för kodgenerering varierar också för olika versioner av samma tekniska plattform.

Att uttömmande beskriva hur kodgenerering går till oavsett teknisk plattform och version låter sig därför inte göras. Istället ges i det här appendixet en översiktlig beskrivning av kodgenerering för två vanliga tekniska plattformar (Java och .NET). Kodgenerering för SSBTEN, SSBTPR och SSBTRO ska fungera på motsvarande sätt, även om bara SSBTGU använts som exempel i det här appendixet.

5.1 Kodgenerering Java

För javaplattformen finns flera möjliga val för generering av JAX-WS-kod från WSDL. Två vanliga alternativ är:

- wsdl2java (Apache CXF).
- wsimport (del av JDK sedan Java 6).

Gemensamt för båda är lösningarna är att xjc (del av JDK sedan Java 6) används för att generera JAXB-klasser från de XML-schemans som refereras i WSDL-filen.

I följande exempel på kodgenerering för javaplattformen används ett javaprojekt med Maven-standardlayout där WSDL- och XML-schemafilerna läggs under `src/main/resources`:

```
<projektrot>
+-- src
|   +-- main
|       +-- resources
|           +-- Ssbtgu.wsdl
|           +-- schema
|               +-- ssbt
|                   |   +-- *.xsd (gemensamma XML-scheman för SSBTGU, SSBTEN, SSBTPR och SSBTRO)
|                   +-- ssbtgu
|                   |   +-- *.xsd (specifika XML-scheman för SSBTGU-tjänsten)
```

Med dessa förutsättningar kan kod genereras från WSDL-fil och XML-scheman genom att köra `wSDL2java` med följande argument stående i projektrotten:

```
wSDL2java -d target/generated-sources/cxf -encoding utf-8
-wSDLLocation classpath:Ssbtgu.wsdl src/main/resources/Ssbtgu.wsdl
```

Samma sak kan också åstadkommas genom att använda Maven och `cxf-codegen-plugin`:

```
<plugin>
<groupId>org.apache.cxf</groupId>
<artifactId>cxf-codegen-plugin</artifactId>
<version>2.7.15</version>
<executions>
<execution>
<id>generate-sources</id>
<phase>generate-sources</phase>
<goals>
<goal>wSDL2java</goal>
</goals>
<configuration>
<sourceRoot>${project.build.directory}/generated-sources/cxf</sourceRoot>
<wSDLOptions>
<wSDLOption>
<wSDL>${project.basedir}/src/main/resources/Ssbtgu.wsdl</wSDL>
<wSDLLocation>classpath:Ssbtgu.wsdl</wSDLLocation>
</wSDLOption>
</wSDLOptions>
</configuration>
</execution>
</executions>
</plugin>
```

Båda genereringsalternativen ovan skapar källkodsfiler under `target/generated-sources/cxf`. Källkodsfilerna läggs i en katalogstruktur som baseras på WSDL-filens och XML-schemans namnrömd. Källkodsfilerna kompileras sedan till klassfiler under `target/classes`. Notera också att det är viktigt att `Ssbtgu.wsdl` och schema-katalogen med alla XML-schemafilerna också följer med i `target/classes`, annars kommer producentanslutningen att exponera en inkorrekt WSDL som genererats från implementationen istället för den riktiga `Ssbtgu.wsdl`.

```

<projektrot>
+-- src
|   +-- main
|       +-- resources
|           +-- Ssbtgu.wsdl
|           +-- schema
|               +-- ssbt
|                   | +-- *.xsd (gemensamma XML-scheman för SSBTGU, SSBTEN, SSBTPR och SSBTRO)
|                   +-- ssbtgu
|                       | +-- *.xsd (specifika XML-scheman för SSBTGU-tjänsten)
+-- target
|   +-- generated-sources
|       | +-- cxf
|       |     +-- se
|       |         +-- bolagsverket
|       |             +-- schema
|       |                 +-- ssbt
|       |                     | +-- **/*.java (genererad kod från gemensamma XML-scheman)
|       |                     +-- ssbtgu
|       |                         +-- v2
|       |                             +-- SsbtguService.java
|       |                             +-- ...
|       |                             +-- grundlaggandeuppgifter
|       |                                 | +-- GrundlaggandeUppgifterBegarar.java
|       |                                 |     ...
|       |                                 +-- ...
|   +-- classes
|       +-- Ssbtgu.wsdl
|       +-- schema
|           | +-- ssbt
|           | | +-- *.xsd
|           | +-- ssbtgu
|           | | +-- *.xsd
|       +-- se
|           +-- bolagsverket
|               +-- schema
|                   +-- ssbt
|                       | +-- **/*.class
|                   +-- ssbtgu
|                       +-- v2
|                           +-- SsbtguService.class
|                           +-- ...
|                           +-- grundlaggandeuppgifter
|                               | +-- GrundlaggandeUppgifterBegarar.class
|                               |     ...
|                               +-- ...

```

Med denna utgångspunkt kan en dataproducent nu att skapa en ”JAX-WS Webservice” för att kunna svara på begäran om grundläggande uppgifter från SSBTGU. Här antas att vi skapat en sådan klass med namnet ProducentanslutningService:

```
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebResult;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
import javax.xml.ws.BindingType;

import se.bolagsverket.schema.ssbtku.v2.SsbtkuService;
import
se.bolagsverket.schema.ssbtku.v2.grundlaggandeuppgifter.GrundlaggandeUppgifterBegaran;
import
se.bolagsverket.schema.ssbtku.v2.grundlaggandeuppgifter.GrundlaggandeUppgifterSvar;

@WebService(wsdlLocation = "Ssbtku.wsdl", targetNamespace =
"http://schema.bolagsverket.se/ssbtgu/v2", name = "SsbtkuService", serviceName =
"SsbtkuServicePorts", portName = "SsbtkuService")
@SOAPBinding(parameterStyle = SOAPBinding.ParameterStyle.BARE)
@BindingType(javax.xml.ws.soap.SOAPBinding.SOAP12HTTP_BINDING)
public class ProducentanslutningService implements SsbtkuService {

    @Override
    @WebResult(name = "GrundlaggandeUppgifter", targetNamespace =
        "http://schema.bolagsverket.se/ssbtgu/v2/grundlaggandeuppgifter",
        partName = "Message")
    @WebMethod(operationName = "GetForetagsuppgifter",
        action = "urn:GetForetagsuppgifter")
    public GrundlaggandeUppgifterSvar getForetagsuppgifter(
        @WebParam(partName = "Message",
            name = "GrundlaggandeUppgifterBegaran", targetNamespace =
                "http://schema.bolagsverket.se/ssbtgu/v2/grundlaggandeuppgifter")
            GrundlaggandeUppgifterBegaran gub)
        throws ServiceException {
        GrundlaggandeUppgifterSvar gus =
            new GrundlaggandeUppgifterSvar();

        // Implementera producentanslutningen här.

        return gus;
    }
}
```

5.2 Kodgenerering .NET

För .NET-plattformen utgår den här beskrivningen från att Visual Studio och C# används. I följande exempel på kodgenerering används ett ”WCF Service Application”-projekt där WSDL och XML-schemafiler lagts direkt under projektrotten:

```
<projektrot>
+-- Ssbtku.wsdl
+-- schema
|   +-- ssbt
|   |   +-- *.xsd (gemensamma XML-scheman för SSBTGU, SSBTEN, SSBTPR och SSBTRO)
|   +-- ssbtgu
|       +-- *.xsd (specifika XML-scheman för SSBTGU-tjänsten)
```

Med dessa förutsättningar kan kod genereras från WSDL-fil och XML-scheman genom att köra svcutil.exe med följande argument stående i projektrotten:

```
svcutil /syncOnly /UseSerializerForFaults /serviceContract /messageContract
/namespace:*,Ssbtku /out:Ssbtku.cs Ssbtku.wsdl schema\ssbtgu\*.xsd schema\ssbt\*.xsd
```


Ovanstående skapar en källkodsfil Ssbtgu.cs där alla genererade klasser ligger i C#-namnrymden Ssbtgu:

```
<projektrot>
+-- Ssbtgu.wsdl
+-- schema
|   +-- ssbt
|       |   +-- *.xsd (gemensamma XML-scheman för SSBTGU, SSBTEN, SSBTPR och SSBTRO)
|       |   +-- ssbtgu
|       |   +-- *.xsd (specifika XML-scheman för SSBTGU-tjänsten)
+-- Ssbtgu.cs (genererad kod från gemensamma och specifika XML-scheman)
```

För att producentanslutningen ska exponera en korrekt WSDL mot SSBTGU krävs ytterligare ett par handgrepp, annars kommer producentanslutningen exponera en inkorrekt WSDL som genererats från implementationen istället för den riktiga Ssbtgu.wsdl:

- WSDL och XML-scheman följer med i publiceringen. I Visual Studio kan detta till exempel åstadkommas genom att sätta propertyn ”Copy to Output Directory” till ”Copy always” för Ssbtgu.wsdl och alla xsd-filer under schema-katalogen.
- Den Ssbtgu.wsdl som följer med i publiceringen måste sedan pekats ut med hjälp av `externalMetadataLocation` i projektets `Web.config`:

```
<behaviors>
  <serviceBehaviors>
    <behavior>
      <serviceMetadata httpGetEnabled="true"
        httpsGetEnabled="true"
        externalMetadataLocation="../Ssbtgu.wsdl"/>
    </behavior>
  </serviceBehaviors>
</behaviors>
```

Ovanstående ska resultera i följande katalogstruktur:

```
<projektrot>
+-- Ssbtgu.wsdll
+-- schema
|   +-- ssbt
|   |   +-- *.xsd
|   +-- ssbtgu
|       +-- *.xsd
+-- Ssbtgu.cs
+-- bin
    +-- Ssbtgu.wsdll
    +-- schema
        +-- ssbt
        |   +-- *.xsd
        +-- ssbtgu
            +-- *.xsd
        +-- <projektrot>.dll
```

Med denna utgångspunkt går det nu skapa en ”WCF Service” för att kunna svara på begäran om grundläggande uppgifter från SSBTGU. Här antas att vi skapat en sådan ”WCF Service” med namnet ProducentanslutningService.svc i projektroten:

```
using Ssbtgu;

namespace Producentanslutning
{
    public class ProducentanslutningService : SsbtguService
    {
        public GetForetagsuppgifterResponse
        GetForetagsuppgifter(GetForetagsuppgifterRequest request)
        {
            var gub = request.GrundlaggandeUppgifterBegaran;
            var gus = new grundlaggandeUppgifterSvar();

            // Implementera producentanslutningen här.

            var response = new GetForetagsuppgifterResponse();
            response.GrundlaggandeUppgifterSvar = gus;
            return response;
        }
    }
}
```