



# XML-dokumentation

För Projektledare & utvecklare  
hos IT-leverantörer till Svenska Intensivvårdsregistret



---

## Innehåll

1.	Om .....	3
2.	XML-dokumentation.....	4
	Innehåll .....	4
	XML-dokumentation.....	4
	Ändringshistorik.....	4
	Exempel .....	4
	XSD.....	4
3.	XML-valideringsklient .....	5
	Transformerering .....	5
4.	NuGet-paket.....	6
	Extensions.....	6



---

## 1. Om

Intensivvårdsavdelningar runt om i landet rapporterar in data till SIR genom att skicka in formaterade XML-filer. Filerna som skickas in går då igenom en validering så att endast korrekt data i korrekt format läses in.

För att förenkla för IT-leverantörer som skapar dessa XML-filer finns detta XML-format tydligt beskrivet i en Excel-fil och det finns även data-klasser för .NET att hämta ned via NuGet.

Vid frågor kontakta SIR. Kontaktuppgifter hittar ni i sidfoten.

---

## 2. XML-dokumentation

En Excel-fil från SIR som på ett tydligt sätt beskriver data-strukturen för en XML-fil. Filen innehåller några olika flikar: Innehåll, XML-dokumentation, Ändringshistorik, Exempel, XSD.

### **Innehåll**

Innehållsförteckning över alla dataklasser inkl. en övergripande beskrivning.

### **XML-dokumentation**

Listar alla dataklasser, dess variabler och alla datatyper anpassade för SIR. En övergripande beskrivningstext finns för alla klasser och variabler. För alla variabler kan ni se vilka värden som är godkända samt vilka variabler som värdet kan utelämnas om så önskas.

### **Automatisk datakomplettering**

Om det finns automatisk datakomplettering för någon variabel i dataklassen så finns det en ruta där automationerna är beskrivna.

### **Valideringsregler**

För varje dataklass finns också en ruta med alla valideringsregler som berör denna klass. Om en regel för en variabel har ändrats så finns det en genväg till just den regeländringen, den hittar ni på variabelns rad längst till höger.

### **Ändringshistorik**

Här finns tydlig ändringshistorik så att ni kontinuerligt kan se vilka ändringar ni behöver anpassa era system efter.

### **Exempel**

Några exempel på hur XML-filer med korrekt XML-struktur kan se ut.

### **XSD**

*XML Schema Definition*

Här finns XSD för alla dataklasser om ni har behov av att exportera dem till ett annat språk.

### 3. XML-valideringsklient

Ett smidigt gränssnitt för validering av XML-filer, hämtas på [xmlvalidering.icuregswe.org](http://xmlvalidering.icuregswe.org).

Välj den fil ni vill validera. När filen är vald kan ni se vilken avdelning filen tillhör, vilken period filen gäller samt antalet vårdtillfällen som finns i filen. Ni kan även se avdelningens inställningar för NEMS, VTS, moduler & åtgärder genom att klicka på *Visa avdelningens inställningar*.

Ni kan välja vilken typ av meddelanden ni vill se från valideringen samt vilken typ av data ni vill se i meddelandetabellen. Inställningarna hittar ni via *kugghjulet* i det övre högra hörnet.

Filvalideringen tar ett par sekunder, detta beror på hur stor XML-filen är. Under tiden filen valideras fylls det på med meddelanden i tabellen, och i det nedre vänstra hörnet kan ni alltid se vilket delmoment valideringen jobbar med. När allt är klart kommer det upp ett meddelande att valideringen är slutförd. För att se mer information ang. ett meddelande kan ni dubbelklicka på meddelandet. Då öppnas en ny ruta där ni kan se vårdtyp, vårdtillfällets start, idbegrepp m.m.

Letar ni efter något specifikt meddelande i tabellen så finns det möjlighet att söka. Vill ni kopiera alla meddelanden finns det även möjlighet till detta. Båda dessa funktioner hittar ni i det nedre högra hörnet.

#### Transformerering

Om ni vill följa den nya XML-standarden så bör ni främst se över varningarna som kommer från transformeringen av XML-filen. För att förenkla denna process kan ni via inställningarna ställa in att ni endast vill se meddelandena från transformeringen i tabellen.

Läser XML-fil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Transformerar XML-fil	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Skriver transformerad fil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

För att bara se den mest relevanta informationen i tabellen kan ni också välja dessa alternativ.

- Visa modul
- Visa indrag
- Visa olika färger beroende på nivå
- Visa kolumn-rubriker

#### Exempel på varningar och dess betydelse

" [SOFA] (Rad: 104860) 'Nej' är inte ett giltigt värde för elementet SOFA.Dobutamin, ändrat till 'false' "

För modul SOFA, på rad 104860 i XML-filen, är inte längre 'Nej' ett värde som får användas för elementet 'SOFA.Dobutamin' utan numera används istället en boolean, alltså 'false'.

" [Vårddata] (Rad: 170155) Döper om element Vårddata->Inskrivningstid till VårdtillfälletsStart "

För modul Vårddata, på rad 170155 i XML-filen, har variabeln 'Inskrivningstid' bytt namn till 'VårdtillfälletsStart'.

" [PIM3] (Rad: 174708) '1,7' är inte ett giltigt värde för elementet PIM3.Laktat, ändrat till '1.7' "

För modul PIM3, på rad 174708, för elementet 'PIM3.Laktat' används nu punkt som decimaltecken istället för komma.

## 4. NuGet-paket

På NuGet.org finns paketet [SIR.XML](#) som innehåller alla olika SIR-klasser. Dessa klasser stämmer helt överens med strukturen som beskrivs i [XML-dokumentationen](#).

Paketet har stöd för några olika .NET-versioner:

- .NETFramework 4.7
  - .NETFramework 4.6.2
  - .NETStandard 2.0
- (Flexibelt som kan konverteras till andra tekniker)

### Exempel

På [SIRs hemsida](#) finns ett exempel-projekt där ni kan se hur SIR.XML används i praktiken genom att ett SIRData-objekt skapas och från det genereras sedan en XML-fil.

### Extensions

I paketet finns en klass som heter Extensions. Dessa tillägg är skapade för att ni på ett enkelt sätt själva ska kunna skapa och testa era egna XML-filer.

Det finns två metoder, ToXML() och Save().

#### ToXML()

XDocument ToXML([this](#) SIRDataVer52 data)

Serialiserar ett objekt av datatyp SIRData till datatyp XDocument.

#### Save()

void Save([this](#) SIRDataVer52 data, [string](#) fileName)

Serialiserar ett objekt av datatyp SIRData och sparar ner XML-filen till angiven plats & namn.

### Användning

```
{  
    var data = new SIR.XML.Ver52.SIRDataVer52();  
  
    //Fyll objektet med data  
  
    var fileName = @"C:\Temp\TestXML.xml";  
  
    data.Save(fileName);  
}
```