

XML -muotoisen raportin luonti

Anita Mäkinen-Jäntti

- XML -tiedostojen kirjoitusmahdollisuudet
 - Tagsettien käyttäminen
 - Data -vaiheen käyttäminen
- Sähköinen siirtoliike ja XML -tiedoston välityksen tarve
- Käytännön esimerkki XML-muotoisen tiedon luomisesta
 - Datan muoto
 - XML -tiedoston luominen
 - Data -vaiheen muuttujat
 - Makromuuttujat

XML -TIEDOSTOJEN LUONTI

- XML -tiedostoja voidaan luoda SAS:lla
 - Valmiita ods tagsettejä käyttämällä.
 - Proc Templatella itse tehtyjä tagsettejä käyttämällä.
 - Data -vaiheella.
- Lukeminen XML libname engineellä
- Hierarkkisen XML:n tuki on vajaata.

TAGSETTIEN KÄYTTÄMINEN

- Perus tagsetit ovat melko vajaita, omien tagsettien käyttö on perusteltua.
- XBRL tuki on olematonta.
- Omien tagsetin luonti on mahdollista proc templatella.
- Jakelun helppous ja monistettavuus.
 - Hierarkkisuus on mahdollista toteuttaa vain vähäisin osin.

DATA -VAIHE

- 90 -lukulaisuudesta huolimatta toimiva ratkaisu
-> vanhatkin tulosteet saadaan nuorennettua 30 vuotta.
- Datan muokkaus tulostusmuotoon kannattaa erottaa tulostuksesta.
- Tulostus tapahtuu data _null_ - vaiheella ja put -lauseilla.

XML LIBNAME ENGINE

- XML libname engine on kätevä tapa saada luettua XML -tiedostoja.
- Hierarkkisuus toteutettu jollain tasolla.
- Vaatii XML MAP -tiedoston taakseen, MAP voidaan kuitenkin luoda esimerkiksi Scheman perusteella.

- Tiedot lähetettiin aikaisemmin (tilastohistoria käyttöön vuonna 2012) vakuutusyhtiöiden välillä pääosin sähköpostitse excelinä tai pdf:nä.
- Tieto pitää saada välitettyä vastaanottajalle XML -muotoisena (sähköisen tilastohistorian välitys vuoden 2013 alusta lähtien).
- Tulostusosuuteen jouduttiin kirjoittamaan vain muutama rivi lisää koodia.
XML -tiedoston käsittelyyn mainframella ja vientiin DB2:aan XML:änä jouduttiin kirjoittamaan vähän lisää koodia.

- Paperille tulostettu raportti oli jo valmiiksi 'määrämuotoinen'.
- Raportti oli toteutettu data `_null_` -tulostuksella ja tulostuksen muotoilu oli hoidettu taulukko templateilla.
- Raportin otsikkotiedot parametreinä välitettyjen makromuuttujien arvoina.

MIKÄ TEKNIikka VALITTIIN?

- Koska raportissa oli jo käytetty proc templatella toteutettuja taulukkorakenteita, yritettiin luoda omaa tagsettiä -> raportin hierarkkisuus ei ollut teknisesti mahdollista toteuttaa tällä tekniikalla.
- Käytössä olleesta tulostusohjelmasta purettiin pois ods -lauseet ja tulosteet ohjattiin tekstitiedostoon.

MITEN TULOSTUS TAPAHTUU?

- Tulostaminen voidaan suorittaa useamman data -vaiheen aikana, mikäli file -lauseessa käytetään mod -optiota.
- Ensin kirjoitetaan otsikkotiedot

```
put '<ns1:TilastotiedotVastaus xsi:schemaLocation = "https://ws.tvl.fi/2012/11/16/vakuutustensiirto/TapvTilastotiedotVastaus  
TapvTilastotiedotVastaus.xsd"
```

```
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns:ns1 = "https://ws.tvl.fi/2012/11/16/vakuutustensiirto/TapvTilastotiedotVastaus"
```

```
  xmlns:Vakio = "https://ws.tvl.fi/2012/11/16/vakuutustensiirto/VakioOsa"
```

```
  xmlns:YleisetTyypit = "https://ws.tvl.fi/2012/11/16/vakuutustensiirto/YleisetTyypit">";
```

```
put " <ns1:Vakio>";
```

```
put "          <Vakio:VastaanottavaLaitos>&vastaanottavalaitos</Vakio:VastaanottavaLaitos>";
```

```
put "          <Vakio:LahettavaLaitos>&lahettavalaitos</Vakio:LahettavaLaitos>";
```

MITEN TULOSTUS TAPAHTUU?

- Kun data on valmis (ja sopivasti nimetty) on tulostaminen helppoa. Tagit kirjoitetaan datan arvojen ympärille:

```
put "<ns1:OhimenevatKorvauksetSuoritusvuodelta><ns1:OhimenevatKorv>";  
put ohim_korvaus_sum;  
put "</ns1:OhimenevatKorv><ns1:Paivarahaa>";  
put paivarahaa_sum;  
put "</ns1:Paivarahaa></ns1:OhimenevatKorvauksetSuoritusvuodelta>";
```

XML -TEKSTITIEDOSTOA

```
<ns1:TilastotiedotVastaus xsi:schemaLocation=
"https://ws.tvl.fi/2012/11/16/vakuutustensiirto/TapvTilastotiedotVastaus TapvTilastotiedotVastaus.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns1="https://ws.tvl.fi/2012/11/16/vakuutustensiirto/TapvTilastotiedotVastaus"
xmlns:Vakio="https://ws.tvl.fi/2012/11/16/vakuutustensiirto/VakioOsa"
xmlns:YleisetTyypit="https://ws.tvl.fi/2012/11/16/vakuutustensiirto/YleisetTyypit">
  <ns1:Vakio>
    <vakio:VastaanottavaLaitos>1</vakio:VastaanottavaLaitos>
    <vakio:LahettavaLaitos>2</vakio:LahettavaLaitos>

  <ns1:OhimenevatKorvauksetSuoritusvuodelta><ns1:OhimenevatKorv>
0
</ns1:OhimenevatKorv><ns1:Paivarahaa>
0
</ns1:Paivarahaa></ns1:OhimenevatKorvauksetSuoritusvuodelta>
```

- Jos itse tulostus on koostunut useammasta kutsutusta osasta, joita on toistettu vaihtelevia määriä, on aloitus ja lopetustagien määrän kohdalleen saaminen tarkkaa.
- XML tiedoston vienti DB2 XML -sarakkeeseen onnistui vain 'Cobol-modulilla' sillä SAS tai Java eivät pystyneet viemään kuin 32K pituisia palasia.

- Jos kannassa on pidempi muuttuja, kuin mitä XML -tiedostossa on määritelty joudutaan muuttujan pituus lyhentämään.

```
put "<YleisetTyypit:AmmattiNimike>" substr(left(ammnimi), 1, 50) "</YleisetTyypit:AmmattiNimike>";
```

```
put "<YleisetTyypit:Palkka>" palkkasumma "</YleisetTyypit:Palkka>";
```

```
put "<YleisetTyypit:PaivienYhteisMaara>0</YleisetTyypit:PaivienYhteisMaara>";
```