



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

FINVOICESENDERINFO- SANOMIEN MUODOSTUSOHJELMA

Opinnäytetyö

TEKIJÄ: Aku Parviainen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Aku Parviainen	
Työn nimi FinvoiceSenderInfo-sanomien muodostusohjelma	
Päiväys 11.5.2014	Sivumäärä/Liitteet 23 / 1
Ohjaaja(t) lehtori Jussi Koistinen	
Toimeksiantaja/Yhteistyökumppani(t) Enfo Zender Oy	
Tiivistelmä <p>Opinnäytetyön aiheena oli FinvoiceSenderInfo-sanomien muodostaminen ohjelmallisesti. Sanomilla laskuttaja voi ilmoittaa tarjoavansa e-laskua verkkopankkiasiakkaille. Laskuttaja lähettää sanomat palveluntarjoajansa (kuten Enfo Zender Oy:n) välityksellä.</p> <p>SenderInfo-sanomat täytyy muodostaa ja lähettää erikseen jokaiseen verkkopankkiin, jossa e-laskua halutaan tarjota. Lisäksi jokaista laskutettavaa tuotetta tai palvelua kohden täytyy muodostaa oma sanoma, joten pelkästään yhdelle laskuttajalle luotavien sanomien määrä voi kasvaa huomattavaksi. Sanomien muodostaminen käsin on aikaa vievää ja virhealtista, joten ratkaisuna ongelmaan päätettiin muodostaa sanomat ohjelmallisesti. Ohjelman kehitykseen käytettiin Visual Studio -kehitysympäristöä ja .NET Framework -komponentteja. Ohjelmointikielenä oli C#.</p> <p>Lopputuloksena toteutettiin ohjelma, jonka avulla käyttäjä voi luoda sanomat ohjelmaan syötteenä annettavien laskuttajatietojen perusteella. Muodostamalla sanomat ohjelmallisesti pystyttiin vähentämään virheiden määrää ja sanomien tekemiseen käytettävää aikaa.</p>	
Avainsanat Finvoice, SenderInfo, Visual Studio, .NET Framework, C#	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Aku Parviainen			
Title of Thesis Application for Creating FinvoiceSenderInfo Messages			
Date	11 May 2014	Pages/Appendices	23 / 1
Supervisor(s) Mr Jussi Koistinen, Lecturer			
Client Organisation /Partners Enfo Zender Oy			
<p>Abstract</p> <p>The purpose of this thesis was to create an application for creating FinvoiceSenderInfo messages. With a SenderInfo message a biller can inform online bank services that it wishes to offer e-invoicing to its customers. The biller sends the messages through a service provider, such as Enfo Zender Oy.</p> <p>SenderInfo messages must be created and send individually for each online bank service where the biller wishes to offer e-invoicing. The messages must also be created for each billable service or product. Because of this the amount of messages that need to be created can be huge. Creating all these messages manually would be time-consuming and error-prone. The solution was to create the messages programmatically. The application was developed with the Visual Studio integrated development environment and by using .NET Framework components. The programming language was C#.</p> <p>As a result of this thesis an application was developed to handle the formation of the messages. The application helped to reduce the amount of errors and time needed to create the messages.</p>			
Keywords Finvoice, SenderInfo, Visual Studio, .NET Framework, C#			

ESIPUHE

Haluan kiittää toimeksiantajaa Enfo Zender Oy:tä mahdollisuudesta opinnäytetyön tekemiseen yhtiölle. Haluan kiittää myös esimiestäni Petteri Saariota ja kaikkia ohjelman käyttäjiä, joilta sain paljon hyvää palautetta kehityksen aikana ja sen jälkeen. Toivon, että ohjelmasta on teille paljon hyötyä.

Lisäksi haluan kiittää opinnäytetyöni ohjaajaa, lehtori Jussi Koistista työn tarkastamisesta ja ohjauksesta.

Kuopiossa 11.5.2014

Aku Parviainen

SISÄLTÖ

TERMIT JA LYHENTEET	6
1 JOHDANTO	7
2 FINVOICE	8
3 VAATIMUKSET	9
4 KÄYTETYT TYÖKALUT JA TEKNIIKAT	10
4.1 C#	10
4.2 Visual Studio Express	10
4.3 Windows Forms	11
4.4 LINQ	11
4.5 GIT	12
5 OHJELMA.....	13
5.1 Lomakemuotoinen muodostus	13
5.2 CSV-muotoinen muodostus.....	14
5.3 Virheiden tarkastus	14
5.4 Asetukset	15
6 TOTEUTUS.....	16
6.1 Määrittely.....	16
6.2 Suunnittelu.....	16
6.3 Ohjelmointi.....	16
6.3.1 Käyttöliittymä.....	16
6.3.2 Tietojen lukeminen.....	17
6.3.3 Tietojen tarkastus	17
6.3.4 Sanomien kirjoittaminen	18
6.4 Testaus	18
6.5 Ongelmat	18
7 YHTEENVETO.....	19
LÄHTEET	20
LIITE 1: FINVOICESENDERINFO MALLISANOMA	21

TERMIT JA LYHENTEET

CSV	Comma-separated values, yksinkertainen tiedostoformaatti, jossa arvot on eroteltu toisistaan pilkulla. Myös muita merkkejä voidaan käyttää erottimena, esim. puolipistettä.
IDE	Integrated development environment, ohjelma tai joukko ohjelmia, joiden avulla ohjelmoija voi suunnitella ja rakentaa sovelluksia.
.NET Framework	Microsoftin kehittämä ohjelmistokomponenttikirjasto.
SOAP	Tietoliikenneprotokolla rakenteisen tiedon välitykseen. Noudattaa XML-syntaksia.
XML	Extensible Markup Language, merkintäkieli, jolla voidaan kuvata ja jäsennellä tietoja sovitun standardin mukaan.

1 JOHDANTO

Opinnäytetyön aiheena on tehdä toimeksiantajalle työkaluohjelma yrityksen sisäiseen käyttöön. Projektin toimeksiantaja on Enfo Zender Oy ja työn aiheena FinvoiceSenderInfo-sanomien muodostaminen ohjelmallisesti. Ohjelman tarkoitus on muodostaa Finvoice-rajapinnan mukaisia SenderInfo-sanomia. SenderInfo-sanomilla laskuttaja voi mm. ilmoittaa tarjoavansa e-laskua verkkopankkiasiakkailla. Laskuttaja lähettää sanomat palveluntarjoajansa (tässä tapauksessa Enfo Zender Oy) välityksellä.

SenderInfo-sanomat sisältävät tietoa laskuttajasta ja laskutuskohteesta. Sanomilla voidaan e-laskupalvelun avauksen lisäksi muokata laskuttajan tietoja tai poistaa palvelu käytöstä. Sanomien muodostaminen käsin on aikaa vievää ja virhealtista. Lisäksi koska yhdellä laskuttajalla voi olla useita eri laskutuskohteita ja koska sanomat täytyy muodostaa erikseen jokaiseen verkkopankkiin, jossa palvelua halutaan tarjota, voi muodostettavien sanomien määrä kasvaa suureksi. Ratkaisuna tähän ongelmaan tehtiin ohjelma, johon voidaan ladata tai syöttää laskuttajan tietoja. Näiden perusteella luodaan sanomat niihin verkkopankkeihin, joissa laskuttaja haluaa tarjota e-laskua.

Toimeksiantaja Enfo Zender Oy on Enfo Oyj:n itsenäinen tytäryhtiö. Uusi yhtiö syntyi, kun Enfo Oyj:n Tiedonvälityspalvelut yhtiöitettiin 1.10.2012. Enfo Zender Oy:n toimitusjohtajana toimii Tero Saksman. Enfo Zender tarjoaa yrityksille tiedonjalostusta ja tiedonvälityspalveluita ja on erityisesti tunnettu laskutusoperaattorina. Enfo Oyj on pohjoismainen IT-palvelutalo, joka työllistää lähes 800 työntekijää Suomessa ja Ruotsissa. (Enfo Oyj 2012.)

2 FINVOICE

Finvoice-sanoma on esitystapa sähköisille laskuille ja muille liiketoimintasanomille. FInvoice-standardin kuvauksen toimittaa Finanssialan Keskusliitto. Finanssialan Keskusliitto on finanssialan yhtiöiden toimialajärjestö, joka edustaa mm. Suomessa toimivia pankkeja (Finanssialan Keskusliitto).

Palveluntarjoajat ja verkkopankit voivat tarjota FInvoice-välityspalvelua noudattamalla FInvoice-standardia. Tähän kuuluu rajapinnan tekninen määrittely, jonka perusteella sanomia voidaan muodostaa ja tulkita. Tyypillisimpiä FInvoice-välityspalvelun kautta lähetettäviä sanomia ovat kuluttajan e-lasku ja suoramaksu, mutta myös yritysten välinen laskutus ja muut sähköiset sanomat, kuten tilaukset ja tarjoukset ovat mahdollisia. FInvoice-sanomien välityspalvelussa sekä lähettäjän että vastaanottajan tulee tukea FInvoice-esitystapaa. FInvoice-sanomat noudattavat XML-syntaksia ja sanomaan kuuluu aina myös siirtokehys (SOAP). (FInvoice 2013.)

FInvoice ilmoittamispalvelun sanomien avulla voidaan mm. välittää laskuttaja- ja vastaanottajatietoja laskuttajan ja maksamisen verkkopalvelun välillä. Ilmoittamispalvelun sanoma voi olla tyyppiä SenderInfo, ReceiverInfo tai ReceiverProposal. SenderInfo-sanomalla laskuttaja voi ilmoittaa tarjoavansa e-laskua verkkopankkiasiakkaille sekä automaattisen maksamisen palvelua, ts. suoramaksua niille asiakkaille, jotka eivät käytä verkkopankkia. ReceiverInfo-sanomalla laskun vastaanottajan (asiakkaan) tiedot välitetään SenderInfo-sanoman lähettäjälle (laskuttajalle). ReceiverProposal-sanomaa käytettiin muuttamaan päättynyt suoraveloituspalvelu joko e-laskuksi tai suoramaksuksi. (FInvoice 2012.)

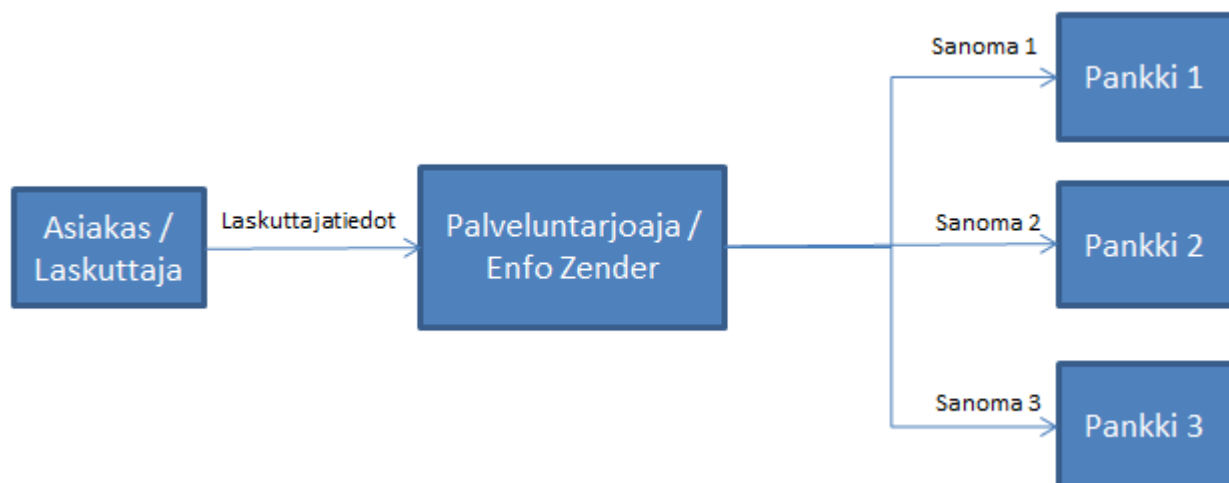
Tässä opinnäytetyössä ja sen lopputuloksena syntyneessä ohjelmassa käsitellään vain SenderInfo-sanomia. SenderInfo-sanomalla laskuttaja välittää omat laskuttajatietonsa maksamisen verkkopalveluun, jossa ne voidaan asettaa asiakkaan käytettäväksi. SenderInfo voi olla tyyppiä ADD, CHANGE tai DELETE. ADD-tunnusta käytetään, kun halutaan tarjota e-laskupalvelua, CHANGE-tunnusta kun halutaan muuttaa aikaisemmin ilmoitettuja tietoja ja DELETE-tunnusta kun halutaan poistaa palvelu käytöstä.

FInvoice-skeemakuvaus, soveltamisohjeet yms. aineisto on vapaasti saatavilla Finanssialan Keskusliiton internet-sivuilta (FInvoice-verkkolasku).

3 VAATIMUKSET

Ohjelman avulla käyttäjän tulee pystyä luomaan Finvoice 2.01 -rajapinnan mukaisia SenderInfo-sanomia. Vaatimukset perustuvat Enfon sisäiseen toiminnanohjausjärjestelmään luotuun tikettiin, joka sisältää ohjelman vaatimusmäärittelyn. Toimeksiannon ja vaatimusmäärittelyn ovat tehneet ohjelman loppukäyttäjät eli projektipäälliköt. Vaatimukseen vaikutti myös heidän antamansa palaute ohjelman kehitysvaiheessa.

Asiakas ilmoittaa vaaditut laskuttajatiedot (kuten tilinumerot ja osoitetiedot) palveluntarjoajalle (Enfo Zender), joka luo tietojen perusteella pankkikohtaiset SenderInfo-sanomat ja lähettää ne pankkeihin (kuvio 1). Lisäksi palveluntarjoaja lisää sanomalle muut Finvoice-skeeman mukaan vaaditut tiedot. Palveluntarjoaja vastaa siitä, että sanomat ovat oikein muodostettu ja välittyvät pankkeihin. Tämän ohjelman vaatimukseen ei kuitenkaan kuulu sanomien lähettäminen, vaan pelkästään sanomien muodostaminen.



KUVIO 1. SenderInfo-sanomien välitys.

Ohjelmassa tulee olla mahdollisuus luoda sanomia syöttämällä laskuttajatiedot ohjelman käyttöliittymän kautta. Lisäksi ohjelmaan täytyy pystyä lataamaan vanhoja sanomia pohjiksi, esimerkiksi laskuttajatietojen muutosta varten tai jos samalle laskuttajalle halutaan luoda uusi laskutuskohte. Näin käyttäjän ei tarvitse aina syöttää samoja tietoja uudestaan.

Lisäksi vaatimusmäärittelyyn kuuluu ominaisuus luoda sanomia lataamalla ohjelmaan laskuttaja- ja laskutuskohtetietoja sisältävä CSV-tiedosto. CSV-tiedoston tulee noudattaa ennalta määriteltyä formaattia, jossa vaaditut tiedot ovat aina samassa järjestyksessä. Käyttäjä voi luoda CSV-tiedoston esimerkiksi taulukkolaskentaohjelmalla. Tämän toiminnon avulla voidaan helposti luoda paljon sanomia kerralla, jos esimerkiksi samalle laskuttajalle halutaan avata useita eri laskutuskohteita.

Vastaanottava pankki tarkastaa sanoman ja hylkää sen tarvittaessa. Sanoman tulee noudattaa Finvoice 2.01 -rajapintaa ja sanoman tulee olla muodollisesti oikein. Sanomalla välittyvien tietojen, kuten IBAN-tilinumeron, täytyy olla oikein muodostettuja. Ohjelma kehitetään Microsoft Windows -käyttöjärjestelmälle.

4 KÄYTETYT TYÖKALUT JA TEKNIIKAT

Ohjelman toteuttamiseksi täytyi valita soveltuva kehitysympäristö (IDE). Kehitysympäristön tuli sisältää valmiit työkalut työpöytäsovelluksen käyttöliittymän rakennukseen ja muut kehitysympäristöltä vaadittavat perusominaisuudet, kuten tekstieditorin ja työkalut virheiden etsintään. Kehitysympäristön tuli olla yhteensopiva Microsoft Windows -käyttöjärjestelmän (Windows 7) kanssa.

Vaihtoehtoisiksi karsiutuivat nopeasti Visual Studio ja NetBeans IDE. Kyseiset työkalut täyttivät helposti vaaditut ominaisuudet, ja käytännössä ohjelman olisi voinut toteuttaa kummalla tahansa näistä. Vaadittavat ominaisuudet täyttäviä kehitystyökaluja olisi ollut monia muitakin, mutta vaihtoehtoisiksi valittiin nämä kaksi lähinnä, koska ne olivat entuudestaan tutuimpia. Näin aikaa ei kulu uuden kehitysympäristön opetteluun ja varsinaisen ohjelman kehittämisen voi aloittaa nopeasti.

Näistä kahdesta valittiin Visual Studio, koska se oli tutumpi ja hieman kehittyneempi kehitysympäristö. Lisäksi Visual Studio ja .NET -ympäristö tarjoavat monia hyödyllisiä komponentteja (kuten LINQ) ja Windows-sovelluskehitykseen sopivat työkalut esim. käyttöliittymän rakennukseen.

4.1 C#

Ohjelman lähdekoodi kirjoitettiin C#-ohjelmointikielellä. C# on Microsoftin .NET-ympäristöä varten kehittämä ohjelmointikieli, jonka syntaksi muistuttaa C:tä, C++:aa tai Javaa. C# on olio-ohjelmointikieli, ja se toteuttaa olio-ohjelmoinnin keskeisimmät konseptit, kuten perinnän ja kapseloinnin. (Microsoft Developer Network a.)

Ohjelmointikielen valinta liittyi läheisesti kehitysympäristön valintaan. Koska C# on Microsoftin kehittämä ohjelmointikieli, tarjoaa Visual Studio tälle myös parhaimman tuen. Jos kehitysympäristöksi olisi valittu NetBeans IDE, olisi ohjelmointikieli ollut Java, koska se on yksi NetBeans IDE:n tukemista ohjelmointikielistä (C# ei ole tuettu) (NetBeans IDE.)

4.2 Visual Studio Express

Visual Studio Express on Microsoftin tarjoama ohjelmistokehitysympäristö (IDE). Visual Studio Express on rajoitetumpi tuote täydestä Visual Studio -kehitysympäristöstä, mutta Express-versio mahdollistaa käytön ilman lisenssimaksua. Käyttäkseen Express-versiota käyttäjän tarvitsee luoda vain ilmainen Microsoft-tili. (Visual Studio.)

Ohjelman kehitys aloitettiin Visual C# 2010 Express -versiolla ja se päivitettiin myöhemmin Express 2012 for Windows Desktop -versioon.

4.3 Windows Forms

Windows Forms on Microsoft .NET Framework –komponentti, jonka avulla voidaan luoda graafisia käyttöliittymiä. Yhdessä Visual Studion käyttöliittymän rakentajan kanssa voidaan helposti luoda sovelluksen käyttöliittymä raahaamalla komponentteja hiirellä lomakkeelle, jolloin ympäristö osaa automaattisesti luoda tätä vastaavan ohjelmakoodin. Tämä vähentää käyttöliittymän rakentamiseen tarvittavaa aikaa.

Toinen .NET-ympäristön tarjoama vaihtoehto olisi ollut Windows Formsista uudempi tekniikka Windows Presentation Foundation (WPF). Molemmilla tekniikoilla voidaan rakentaa ohjelman vaatimukset täyttävä käyttöliittymä, mutta WPF soveltuu paremmin ns. modernin käyttöliittymän rakentamiseen (visuaalisuus, animaatiot, mediaelementit). Muitakin eroja löytyy, esimerkiksi WPF:ssä käyttöliittymä kuvataan XAML:n avulla, joka on XML-tyyppinen kuvauskieli. (Microsoft Developer Network b.)

Näistä kahdesta valittiin Windows Forms, koska sen tapa toteuttaa käyttöliittymä oli tutumpi. Lisäksi koska ohjelman käyttöliittymä on hyvin yksinkertainen, ei WPF:n tarjoamille lisäominaisuuksille olisi ollut käyttöä. Windows Forms on myös erittäin tuettu ja tuttu tekniikka, jolla on vastaavanlaisia käyttöliittymiä toteutettu jo pitkän aikaa.

4.4 LINQ

LINQ eli Language-Integrated Query on Microsoft .NET Framework –komponentti, jonka avulla voidaan tehdä kyselyjä esimerkiksi tietokantoihin tai XML-tiedostoihin. Sen syntaksi muistuttaa SQL-kyselykieltä. Tässä ohjelmassa sitä käytettiin mm. SenderInfo XML -sanomien lukemiseen ja kirjoittamiseen. LINQ:n avulla on esimerkiksi helppo luoda uusia XML-elementtejä (kuvio 2 ja kuvio 3).

```
XElement yhteystiedot =
    new XElement("Yhteystiedot",
        new XElement("Yhteystieto",
            new XElement("Nimi", "Matti Meikäläinen"),
            new XElement("Puhelin", "012 345 678 9"),
            new XElement("Osoite",
                new XElement("Katuosoite", "Katu 7"),
                new XElement("Postitoimipaikka", "Kuopio"),
                new XElement("Postinumero", "00000")
            )
        )
    );
```

KUVIO 2. C# LINQ-esimerkki, elementin luominen.

```
<Yhteystiedot>
  <Yhteystieto>
    <Nimi>Matti Meikäläinen</Nimi>
    <Puhelin>012 345 678 9</Puhelin>
    <Osoite>
      <Katuosoite>Katu 7</Katuosoite>
      <Postitoimipaikka>Kuopio</Postitoimipaikka>
      <Postinumero>00000</Postinumero>
    </Osoite>
  </Yhteystieto>
</Yhteystiedot>
```

KUVIO 3. Kuvion 2 LINQ-komennolla luotu XML-elementti.

4.5 GIT

Projektin versionhallintaan käytettiin GIT-versionhallintaohjelmistoa. GIT on ilmainen ja avoimeen lähdekoodiin perustuva, hajautetusti toimiva versionhallintaohjelmisto. Hajautettu toimintaperiaate tarkoittaa, että jokaisella kehittäjällä on oma työversio lähdekoodista (eli paikallinen kopio), joka voidaan yhdistää alkuperäiseen, yleensä palvelimella sijaitsevaan kopioon.

5 OHJELMA

Ohjelmassa on kaksi tapaa luoda SenderInfo-sanomia: lomakemuotoinen ja CSV-muotoinen. Suurin ero näiden kahden tavan välillä on se miten käyttäjä syöttää vaaditut tiedot ohjelmaan. Virheiden tarkastus ja sanomien muodostus toimivat samalla tavalla.

SenderInfo-sanomat voivat olla tyyppiä ADD, CHANGE tai DELETE. Käytännössä lomakemuotoisella muodostuksella luodaan CHANGE- ja DELETE-tyyppisiä ja CSV-muotoisella ADD-tyyppisiä. Ohjelma muodostaa valmiit sanomat ohjelman asetuksista määriteltyn tallennuskansioon. Ohjelmassa ei ole toimintoa valmiiden sanomien siirtämiseen pankkeihin, vaan käyttäjän vastuulle jää siirtää sanomat eteenpäin.

5.1 Lomakemuotoinen muodostus

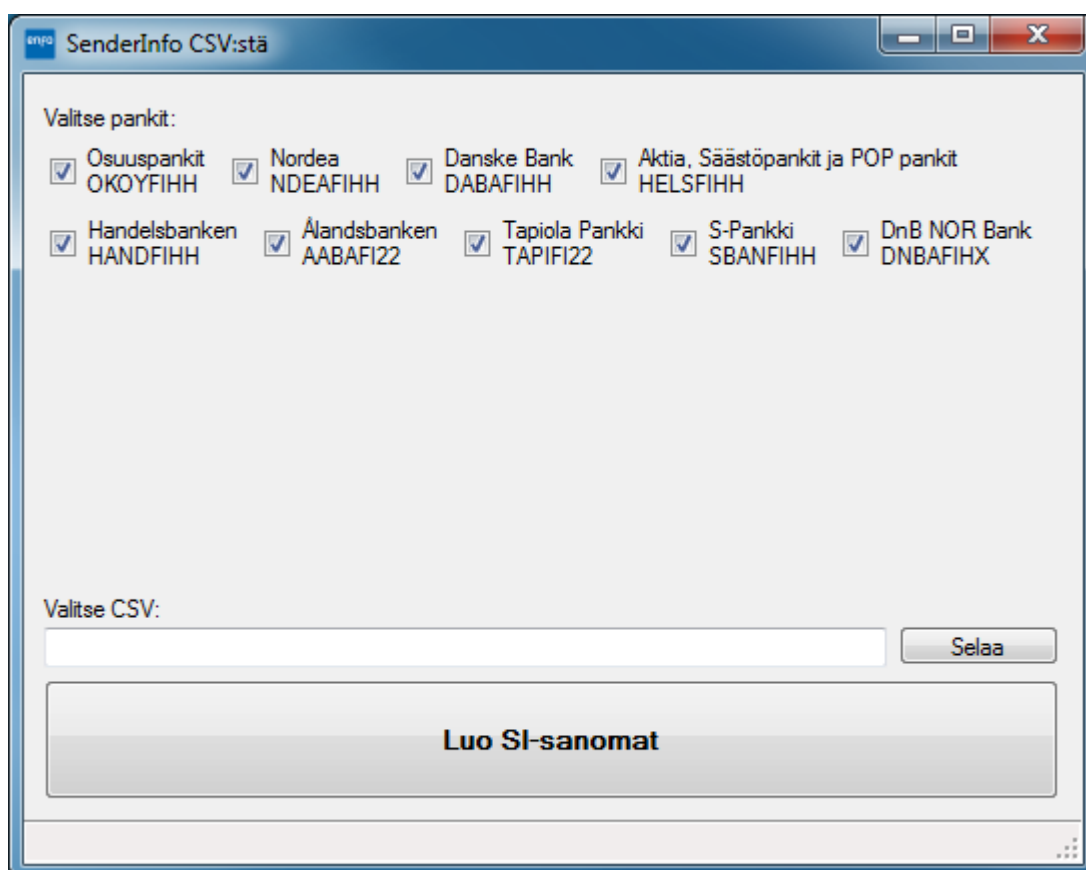
Lomakemuotoisen muodostuksen ensisijainen käyttötarkoitus on luoda sanomia aikaisemmin luotujen sanomien pohjalta. Käyttöliittymän (kuva 1) kautta käyttäjä voi valita SenderInfo-sanoman, josta tiedot luetaan käyttöliittymän kenttiin. On myös mahdollista kirjoittaa kaikki tiedot itse.

KUVA 1. Lomakemuotoinen muodostus.

5.2 CSV-muotoinen muodostus

CSV-muotoisessa muodostuksessa sanomille kirjoitettavat tiedot ladataan CSV-tiedostosta, jonka käyttäjä valitsee ohjelman käyttöliittymän kautta (kuva 2). CSV-tiedoston tulee noudattaa ennalta määritettyä formaattia, jossa tiedot tulevat aina samassa järjestyksessä. Tiedoston ensimmäinen rivi on aina otsikkorivi ja jokaisesta sen jälkeen tulevasta rivistä muodostetaan sanomat käyttöliittymästä valittuihin pankkeihin. Esimerkiksi jos CSV-tiedostossa on otsikkorivin lisäksi 10 riviä ja käyttöliittymästä on valittu 9 pankkia, luodaan $10 * 9 = 90$ sanomaa.

CSV-muotoisessa muodostuksessa on mahdollista luoda vain ADD-tyyppisiä sanomia, ts. lisäysilmoituksia joilla laskuttaja luo uusia laskutuskohteita pankkiin. Tämän ensisijainen käyttötarkoitus on siis kun halutaan luoda kerralla useita uusia laskutuskohteita. CSV-tiedostoa voidaan täyttää helposti esimerkiksi taulukkolaskentaohjelmalla (kuten Microsoft Excel), jonka jälkeen valmis tiedosto ladataan ohjelmaan ja muodostetaan sanomat.

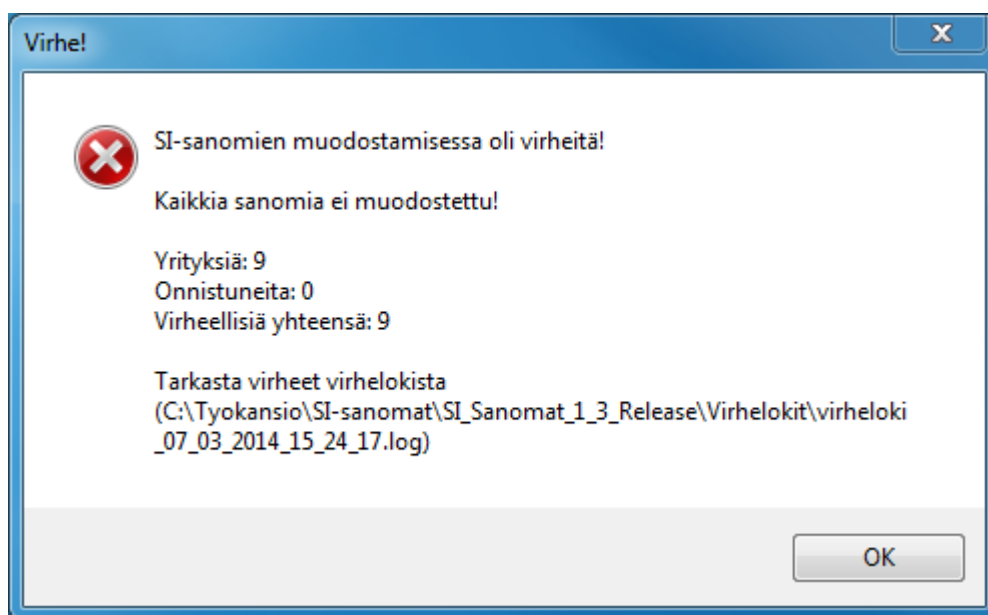


KUVA 2. CSV-muotoinen muodostus.

5.3 Virheiden tarkastus

Ohjelma tarkastaa sanoman tietoja ohjelman oman sisäisen logiikan mukaisesti ja FinvoiceSenderInfo-skeemaa vasten. Ennen tarkastusta muokataan tietoja tarvittaessa oikeaan muotoon, esim. täydennetään puuttuvat etunollat postinumeroon.

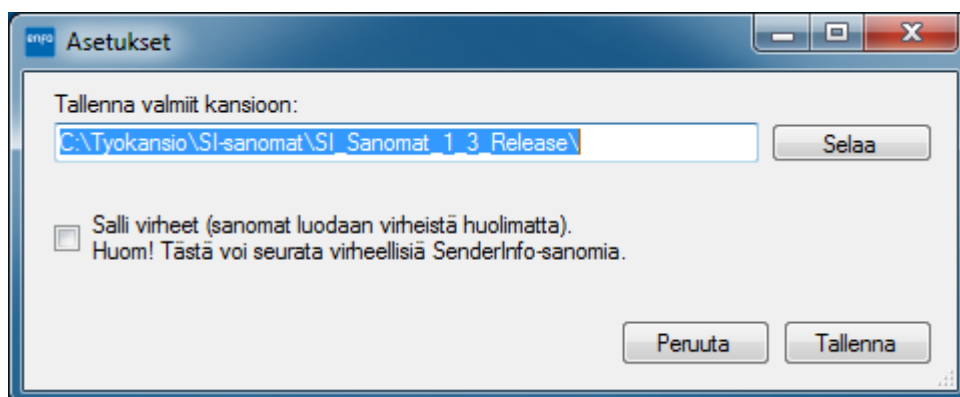
Tarkastettavia tietoja on esimerkiksi tilinumeroiden ja Y-tunnuksen oikeellisuus. Lisäksi tarkastetaan että tietyt pakolliseksi määritellyt tiedot on annettu. Muodostettaessa sanomia ohjelma ilmoittaa jos virheitä löytyi sekä virheellisten sanomien määrän (kuva 3). Tarkemmat virheilmoitukset kirjoitetaan virhelokiin (kuva 5 ja kuva 6).



KUVA 3. Virheilmoitus.

5.4 Asetukset

Ohjelman käyttöliittymän kautta voidaan muokata yleisimpiä asetuksia (kuva 4). Nämä asetukset ovat valmiiden SenderInfo-sanomien tallennuskansio ja mahdollisuus ottaa virheiden tarkastus pois päältä. Kaikkia asetuksia ei voida muokata käyttöliittymän kautta, vaan pelkästään asetustiedostoa käsin muokkaamalla. Nämä asetukset ovat tiettyjä sanomille kirjoitettavia vakiotietoja ja mm. käyttöliittymästä valittavien pankkien tiedot (kuten nimi ja BIC). Näiden tietojen on ajateltu olevan sellaisia joita ei tarvitse muokata tai tarvitsee muokata hyvin harvoin, mutta on haluttu säilyttää mahdollisuus näiden muokkaamiseen ilman ohjelman lähdekoodin muokkausta. Esimerkiksi pankkien tietoja tuskin tarvitsee muokata päivittäisessä käytössä, mutta tämä mahdollisuus on oltava olemassa esimerkiksi jos pankin nimi vaihtuu.



KUVA 4. Asetukset.

6 TOTEUTUS

Työtä tehtiin toimeksiantajan välineillä toimeksiantajan tiloissa Kuopiossa. Toteutuksen aikana pystyttiin hyödyntämään toimeksiantajan palveluksessa työskentelevien henkilöiden teknistä ja Finvoice-rajapintaan liittyvää tietoa. Ohjelman loppukäyttäjien kanssa pidettiin palaveria aina tarpeen niin vaatiessa, mikä helpotti tarkoituksenmukaisen ohjelman kehittämistä.

6.1 Määrittely

Ohjelman vaatimusmäärittely saatiin projektipäälliköiltä. Ohjelman tuli noudattaa Enfollla asetettujen vaatimusten lisäksi Finvoice-rajapintaa muodostettavien sanomien osalta. Määrittelyä päivitettiin yhteistyössä projektipäälliköiden kanssa uusien tarpeiden ja vaatimusten noustessa esille. Alkuperäiset määrittelyt sijaitsevat Enfon työnohjausjärjestelmässä.

6.2 Suunnittelu

Ohjelman tekninen toteutus suunniteltiin määrittelyssä esille tulleiden seikkojen pohjalta. Koska ohjelma ei suoraan liity mihinkään valmiiseen rajapintaan, ei toteutuksessa käytettävien tekniikoiden ja työkalujen valintaa rajoittanut kuin ohjelmistojen mahdolliset lisenssit. Tästä syystä pystyttiin valitsemaan mahdollisimman tutut työkalut, joten aikaa ei mennyt kehitysympäristön opetteluun.

Määrittelyn perusteella hahmotettiin nopeasti kokonaiskuva siitä, kuinka ohjelman tulisi toimia. Suunnittelussa ei kuitenkaan menty kovin yksityiskohtaiselle tasolle, vaan tekniset yksityiskohdat täydentyivät kehitysvaiheessa. Pääpiirteet ja suurin osan käytettävistä tekniikoista tiedettiin, mutta nämäkin tulivat lopulliseen muotoonsa vasta kehityksessä. Lisäksi alkuperäiseen määrittelyyn tuli muutoksia ja täydennyksiä vielä kehitysvaiheessa, mikä aiheutti muutoksia suunniteltuun toteutukseen.

6.3 Ohjelmointi

6.3.1 Käyttöliittymä

Käyttöliittymä rakennettiin Visual Studion käyttöliittymänrakentajalla Windows Forms -komponentteja käyttämällä. Koska yksi tärkeimmistä syistä ohjelman kehittämiseen oli vähentää sanomien muodostamiseen käytettävää aikaa, tuli myös käyttöliittymän olla nopea käyttää. Ohjelman kehityksen aikana käyttöliittymä muuttuikin käyttäjiltä saadun palautteen mukaan, mm. useimmin käytetyt toiminnot tuotiin nopeammin saataville heti aloitusnäkyeseen.

6.3.2 Tietojen lukeminen

Vaikka ohjelmassa voidaan luoda täysin uusia sanomia, sen pääasialliseen tarkoitukseen tarvitaan tietojen lukemista jo muodostetuilta sanomilta tai CSV-tiedostosta. CSV-tiedoston yksinkertaisen rakenteen vuoksi toteutettiin lukeminen tiedostosta itse. Valmiiden SenderInfo-sanomien (XML) lukemiseen käytettiin apuna LINQ-kyselyjä.

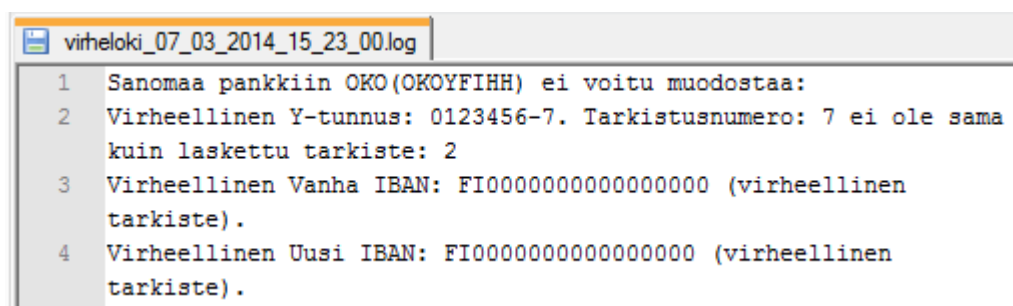
Ohjelmassa on täytynyt ottaa huomioon, että CSV todennäköisesti luodaan tai sitä ainakin muokataan Excel-taulukkolaskentaohjelmalla. Excel saattaa hävittää tiedostoa tallennettaessa esim. etunollia postinumeroista, joten ohjelma täydentää puuttuvia etunollia tarvittaessa.

Ohjelman asetustiedosto noudattaa XML-syntaksia, joten sen lukemisessa hyödynnettiin myös LINQ-kyselyjä.

6.3.3 Tietojen tarkastus

Sanomille kirjoitettava tieto tarkastetaan Finvoice-skeemaa vasten ja ohjelman sisäisen logiikan mukaisesti. Skeemaa vasten voidaan tarkastaa tietoja, kuten onko pakolliset kentät täytetty, mikä on merkkijonojen pituus tai tiedon tyyppi (numeerinen, teksti yms.). Pelkkää skeemaa vasten tarkastamalla ei kuitenkaan voida suorittaa kaikkia vaadittuja tarkastuksia, kuten tilinumeron oikeellisuus tai OVT-tunnuksen tarkastus. Niinpä ohjelmaan täytyi rakentaa tietyt tarkastukset itse.

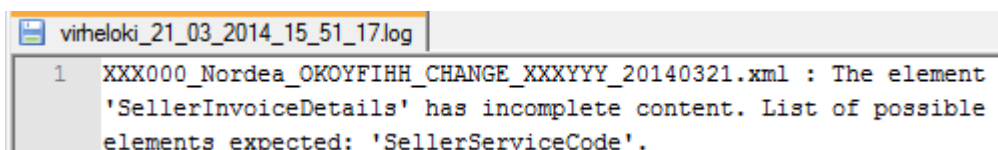
Suurin osa tarkastuksista voitiin kuitenkin tehdä skeemaa vasten vertaamalla. Käyttäjän syöttämät tai tiedostosta lataamat tiedot tarkastetaan ensin ohjelman sisäisen logiikan mukaisesti, ja jos virheitä ei löydy voidaan muodostaa SenderInfo-sanoma. Tässä vaiheessa sanomaa ei vielä kirjoiteta tiedostoon, vaan se tarkastetaan vielä skeematiedostoa vasten. Jos sanomalla olevat tiedot läpäisivät nämä molemmat tarkastukset, voidaan sanoma kirjoittaa tiedostoon. Jos virheitä löytyi, sanomat jätetään kirjoittamatta ja löytyneet virheet kirjoitetaan virhelokiin (kuva 5 ja kuva 6).



```

virheloki_07_03_2014_15_23_00.log
1 Sanomaa pankkiin OKO(OKOYFIHH) ei voitu muodostaa:
2 Virheellinen Y-tunnus: 0123456-7. Tarkistusnumero: 7 ei ole sama
  kuin laskettu tarkiste: 2
3 Virheellinen Vanha IBAN: FI0000000000000000 (virheellinen
  tarkiste).
4 Virheellinen Uusi IBAN: FI0000000000000000 (virheellinen
  tarkiste).
  
```

KUVA 5. Virheilmoitus, ohjelman sisäinen tarkastus.



```

virheloki_21_03_2014_15_51_17.log
1 XXX000_Nordea_OKOYFIHH_CHANGE_XXXYYY_20140321.xml : The element
  'SellerInvoiceDetails' has incomplete content. List of possible
  elements expected: 'SellerServiceCode'.
  
```

KUVA 6. Virheilmoitus, skeemaa vasten tarkastus.

6.3.4 Sanomien kirjoittaminen

Tietojen tarkastuksen jälkeen voidaan virheettömät sanomat kirjoittaa tiedostoon. Tiedostot kirjoitetaan käyttäjän ohjelman asetuksista valitsemaan tallennuskansioon. XML-muotoisten sanomien kirjoitus toteutettiin kirjoittamalla kaikki tarvittavat elementit lähdekoodissa valmiiden LINQ-metodien avulla. Kaikkia Finvoice-skeemassa vapaaehtoiseksi määriteltyjä elementtejä ei kirjoiteta.

6.4 Testaus

Ohjelman muodostamia sanomia pystyttiin testaamaan vertaamalla niitä mallisanomiin. Lisäksi kirjoitettiin yksikkötestit kriittisimmille osille, kuten tilitietojen tarkastukselle. Kehitysvaiheessa testaukseen riitti vain mallisanomaan vertaaminen, mutta myöhemmin ohjelman rakenteen monimutkaistuksessa ja ensimmäisen version käyttöönoton jälkeen lisättiin yksikkötestejä helpottamaan muutosten testausta. Täydellistä kattavuutta ei yksikkötesteillä kuitenkaan rakennettu, osittain koska yksikkötestit lisättiin vasta jälkeempään.

Ohjelma lisättiin myös mukaan Jenkins continuous integration –prosessiin (”jatkuva integraatio”), jonka yhtenä ominaisuutena voidaan aina lisätessä uutta lähdekoodia versionhallintaan kääntää ohjelma ja ajaa määritellyt yksikkötestit. Tällöin huomataan automaattisesti, jos ohjelmassa on virheitä. Nämä ominaisuudet ovat erityisen hyödyllisiä usean kehittäjän ympäristössä, mutta helpottavat testausta myös kehitettäessä itsenäisesti. Jenkins on vapaan lähdekoodin continuous integration -palvelin (Jenkins).

6.5 Ongelmat

Jonkin verran ongelmia aiheutti Finvoice-sanoman muoto, jossa sanomaan kuuluu SOAP-kehys ja varsinainen sisältö. SOAP-kehys lisätään sanoman alkuun niin, että sanomaan tulee XML-standardin vastaisesti kaksi juurielementtiä. Tämä aiheuttaa virheen käytettäessä esimerkiksi LINQ-kyselyjä. Ongelma oli kuitenkin helppo ratkaista erottamalla SOAP-kehys ja varsinainen sisältö toisistaan sanomaa luettaessa ja kirjoittaessa.

7 YHTEENVETO

Opinnäytetyön tavoitteena oli tehdä ohjelma helpottamaan FinvoiceSenderInfo-sanomien muodostusta. Lähetettävien sanomien määrän kasvaessa oli tarve muodostusohjelmalle todellinen. Tapaus oli hyvä esimerkki siitä, kuinka päivittäiseen työhön liittyviä rutiinitehtäviä voidaan nopeuttaa tekemällä ne ohjelmallisesti. Ohjelman avulla pystytään esimerkiksi tekemään sanomat kaikkiin verkkopankkeihin samassa ajassa, mitä ennen meni yhteen sanomaan. Lisäksi voidaan vähentää virheellisten sanomien määrää, kun syötetyt tiedot ja sanoman rakenne tarkastetaan ohjelmallisesti. Tämä ei ollut mahdollista tai käytännöllistä, kun sanomia muodostettiin käsin.

Ohjelman käyttäjiltä saadun palautteen perusteella ohjelmalle asetetut tavoitteet täyttyivät. Hyvään lopputulokseen vaikutti paljon ohjelman käyttäjiltä saadut kehitysideoita ja kommentit. Kun ohjelma otettiin ensimmäistä kertaa käyttöön, nousi esiin vielä uusia tarpeellisia toimintoja, joita ei alun perin osattu ottaa huomioon. Tästä opin nopean käyttöönoton tärkeyden, sillä on miltei mahdotonta ottaa kaikkia asioita huomioon ensimmäisellä kerralla.

Ohjelman suunnitteluun käytetty aika oli melko lyhyt, ja varsinainen toteutus alkoi nopeasti toimeksiannon saatua. Tästä syystä sain perustoiminnot nopeasti valmiiksi, mutta myöhemmin uusien toimintojen myötä jouduin muokkaamaan lähdekoodia enemmän, kuin jos alun perin olisin ottanut uusien toimintojen lisäämisen paremmin huomioon. Ohjelman lähdekoodista ei kovin monimutkaista rakennetta kuitenkaan syntynyt, vaikka projektista tuli hieman laajempi, kuin alun perin olin arvellut.

Asiat, jotka olisin voinut tehdä toisin, liittyvät pieniin teknisiin yksityiskohtiin. Kaiken kaikkiaan olen kuitenkin tyytyväinen lopputulokseen, joka on toimiva ja vaatimukset täyttävä ohjelma.

LÄHTEET

ENFO OYJ 2012. Enfon Tiedonvälityspalvelut yhtiötetään Enfo Zender Oy:ksi [verkkosivu]. [Viitattu 2014-04-17.] Saatavissa: <http://www.enfo.fi/pressrelease/5238/>

FINANSSIALAN KESKUSLIITTO. Perustietoa Finanssialan Keskusliitosta [verkkosivu]. [Viitattu 2014-04-23.] Saatavissa: http://www.fkl.fi/tietoa_meista/Sivut/default.aspx

FINVOICE-VERKKOLASKU. Finvoice - Tekniset tiedostot [verkkosivu]. [Viitattu 2014-04-22.] Saatavissa: http://www.fkl.fi/verkkolasku/yrityksen_verkkolasku/yrityksen_verkkolasku.htm

FINVOICE 2013. Soveltamisohje versio 2.01 [verkkodokumentti]. [Viitattu 2014-04-23.] Saatavissa: http://www.fkl.fi/verkkolasku/yrityksen_verkkolasku/ladattavat/Tekniset%20tiedostot/Finvoice_2_1_soveltamisohje.pdf

FINVOICE 2012. Ilmoittamispalvelun soveltamisohje [verkkodokumentti]. [Viitattu 2014-04-23.] Saatavissa: http://www.fkl.fi/verkkolasku/yrityksen_verkkolasku/ladattavat/Tekniset%20tiedostot/ilmoittamispalvelu/Ilmoittamispalvelun_soveltamisohje_20032012.pdf

JENKINS. An extendable open source continuous integration server [verkkosivu]. [Viitattu 2014-04-22.] Saatavissa: <http://jenkins-ci.org/>

MICROSOFT DEVELOPER NETWORK a. Introduction to the C# Language and the .NET Framework [verkkosivu]. [Viitattu 2014-04-13.] Saatavissa: <http://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>

MICROSOFT DEVELOPER NETWORK b. Introduction to WPF [verkkosivu]. [Viitattu 2014-04-13.] Saatavissa: [http://msdn.microsoft.com/en-us/library/aa970268\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx)

NETBEANS IDE. NetBeans IDE Features [verkkosivu]. [Viitattu 2014-04-13.] Saatavissa: <https://netbeans.org/features/index.html>

VISUAL STUDIO. Visual Studio Express [verkkosivu]. [Viitattu 2014-04-22.] Saatavissa: <http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>

LIITE 1: FINVOICESENDERINFO MALLISANOMA

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:eb="http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ envelope.xsd">
<SOAP-ENV:Header>
<eb:MessageHeader xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
msg-header-2_0.xsd" SOAP-ENV:mustUnderstand="1" eb:version="2.0">
<eb:From>
<eb:PartyId>003712345678</eb:PartyId>
<eb:Role>Sender</eb:Role>
</eb:From>
<eb:From>
<eb:PartyId>NDEAFIHH</eb:PartyId>
<eb:Role>Intermediator</eb:Role>
</eb:From>
<eb:To>
<eb:PartyId>SENDERINFO</eb:PartyId>
<eb:Role>Receiver</eb:Role>
</eb:To>
<eb:To>
<eb:PartyId>OKOYFIHH</eb:PartyId>
<eb:Role>Intermediator</eb:Role>
</eb:To>
<eb:CPAId>yoursandmycpa</eb:CPAId>
<eb:Service>Routing</eb:Service>
<eb:Action>ProcessInvoice</eb:Action>
<eb:MessageData>
<eb:MessageId>XXX000/1</eb:MessageId>
<eb:Timestamp>2014-02-18T07:29:12+02:00</eb:Timestamp>
</eb:MessageData>
</eb:MessageHeader>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<eb:Manifest eb:id="Manifest" eb:version="2.0">
<eb:Reference eb:id="FinvoiceSenderInfo" xlink:href="XXX010">
<eb:Schema
eb:location="http://www.pankkiyhdistys.fi/verkkolasku/finvoice/FinvoiceSenderInfo.xsd"
eb:version="2.0"/>
</eb:Reference>
</eb:Manifest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
<?xml version="1.0" encoding="ISO-8859-15"?>
<?xml-stylesheet type="text/xsl" href="FinvoiceSenderInfo.xsl"?>
<FinvoiceSenderInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FinvoiceSenderInfo.xsd" Version="2.0">
<MessageDetails>
<MessageTypeCode>SENDERINFO</MessageTypeCode>
<MessageTypeText>LASKUTTAJAN ILMOITUS</MessageTypeText>
<MessageActionCode>CHANGE</MessageActionCode>
<MessageDate Format="CCYYMMDD">20140218</MessageDate>
<SenderInfoIdentifier>XXX000</SenderInfoIdentifier>
</MessageDetails>
<SellerPartyDetails>
<SellerPartyIdentifier>0123456-7</SellerPartyIdentifier>
<SellerOrganisationNames LanguageCode="FI">
<SellerOrganisationName>Laskuttaja Oy</SellerOrganisationName>
</SellerOrganisationNames>
<SellerOrganisationNames LanguageCode="SV">

```

```

<SellerOrganisationName>Laskuttaja AB</SellerOrganisationName>
</SellerOrganisationNames>
<SellerOrganisationNames LanguageCode="EN">
<SellerOrganisationName>Laskuttaja LTD</SellerOrganisationName>
</SellerOrganisationNames>
<SellerPostalAddressDetails>
<SellerStreetName>Katutie 1</SellerStreetName>
<SellerTownName>Kökkölä</SellerTownName>
<SellerPostCodeIdentifier>00000</SellerPostCodeIdentifier>
<CountryCode>FI</CountryCode>
<CountryName>Suomi</CountryName>
</SellerPostalAddressDetails>
</SellerPartyDetails>
<InvoiceSenderInformationDetails>
<SellerWebaddressNameText>webbisivut</SellerWebaddressNameText>
<SellerWebaddressText>www.webbisivut.fi</SellerWebaddressText>
<InvoiceSenderAddress>003712345678</InvoiceSenderAddress>
<InvoiceSenderIntermediatorAddress>NDEAFIHH</InvoiceSenderIntermediatorAddress>
</InvoiceSenderInformationDetails>
<SellerAccountDetails>
<SellerAccountID IdentificationSchemeName="IBAN">FI0000000000000000</SellerAccountID>
<SellerBic IdentificationSchemeName="BIC">OKOYFIHH</SellerBic>
<NewSellerAccountID IdentificationSchemeName="IBAN">FI0000000000000000</NewSellerAccountID>
<NewSellerBic IdentificationSchemeName="BIC">OKOYFIHH</NewSellerBic>
</SellerAccountDetails>
<SellerAccountDetails>
<SellerAccountID IdentificationSchemeName="IBAN">FI0000000000000000</SellerAccountID>
<SellerBic IdentificationSchemeName="BIC">NDEAFIHH</SellerBic>
<NewSellerAccountID IdentificationSchemeName="IBAN">FI0000000000000000</NewSellerAccountID>
<NewSellerBic IdentificationSchemeName="BIC">NDEAFIHH</NewSellerBic>
</SellerAccountDetails>
<SellerInvoiceDetails>
<SellerDirectDebitIdentifier>012345678</SellerDirectDebitIdentifier>
<PaymentInstructionIdentifier>XXX000</PaymentInstructionIdentifier>
<SellerInstructionFreeText LanguageCode="FI">Katso asiakasnumero ja sopimusnumero
laskultasi.</SellerInstructionFreeText>
<SellerInstructionFreeText LanguageCode="SV">Titta din kundnummer och avtalsnummer från
fakturan.</SellerInstructionFreeText>
<SellerInstructionFreeText LanguageCode="EN">Check your customer number and agreement number
from your invoice.</SellerInstructionFreeText>
<SellerInvoiceTypeDetails>
<SellerInvoiceTypeText LanguageCode="FI">Lehtitilaus</SellerInvoiceTypeText>
<SellerInvoiceIdentifierText LanguageCode="FI"
SellerInvoiceIdentifierType="01">Asiakasnumero</SellerInvoiceIdentifierText>
<SellerInvoiceIdentifierText LanguageCode="FI"
SellerInvoiceIdentifierType="99">Sopimusnumero</SellerInvoiceIdentifierText>
</SellerInvoiceTypeDetails>
<SellerInvoiceTypeDetails>
<SellerInvoiceTypeText LanguageCode="SV">Tidningsprenumeration</SellerInvoiceTypeText>
<SellerInvoiceIdentifierText LanguageCode="SV"
SellerInvoiceIdentifierType="01">Kundnummer</SellerInvoiceIdentifierText>
<SellerInvoiceIdentifierText LanguageCode="SV"
SellerInvoiceIdentifierType="99">Avtalsnummer</SellerInvoiceIdentifierText>
</SellerInvoiceTypeDetails>
<SellerInvoiceTypeDetails>
<SellerInvoiceTypeText LanguageCode="EN">Subscription</SellerInvoiceTypeText>
<SellerInvoiceIdentifierText LanguageCode="EN" SellerInvoiceIdentifierType="01">Customer
number</SellerInvoiceIdentifierText>
<SellerInvoiceIdentifierText LanguageCode="EN" SellerInvoiceIdentifierType="99">Agreement
number</SellerInvoiceIdentifierText>
</SellerInvoiceTypeDetails>
<SellerServiceCode>01</SellerServiceCode>

```

```
</SellerInvoiceDetails>  
<ProposedDueDateAccepted>NO</ProposedDueDateAccepted>  
<ProposedInvoicePeriodAccepted>NO</ProposedInvoicePeriodAccepted>  
</FinvoiceSenderInfo>
```