

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

Tutkintotyö

Tommi Peräjoki

**OSCAR FOR WINDOWS -TOIMINNANOHJAUSJÄRJESTELMÄN
TOIMITUS- JA ASENNUSJÄRJESTELMÄ**

Työn ohjaaja
Työn teettäjä
Tampere 2007

Ohjelmistotekniikan lehtori Jari Mikkolainen
Oscar Software Oy, valvojana M.Sc Jukka Harkki

TAMPEREEN AMMATTIKORKEAKOULU

Tietotekniikka

Ohjelmistotekniikka

Peräjoki, Tommi

Oscar for Windows -toiminnanohjausjärjestelmän toimitus- ja
asennusjärjestelmä

Tutkintotyö

30 sivua + 23 liitesivua

Työn ohjaaja

Ohjelmistotekniikan lehtori Jari Mikkolainen

Työn teettäjä

Oscar Software Oy, valvojana M.Sc Jukka Harkki

Toukokuu 2007

Hakusanat

ERP, toimitus, asennus

TIIVISTELMÄ

Tässä työssä kerrotaan Oscar Software Oy:n ohjelman Oscar for Windows -toiminnanohjausjärjestelmän uuden toimitus- ja asennusjärjestelmän tekemisen vaiheista sekä käydään läpi nykyisen toimitus- ja asennusjärjestelmän kautta tehtävän toimittamisen ja asentamisen vaiheet Oscar for Windowsille.

Työssä käydään aluksi läpi Oscar Software Oy:n taustaa ja kuvataan nykyisen asennusjärjestelmän rakennetta ja sen sisältämiä ongelmia, josta tämä työkin sai aiheensa. Nykyisen järjestelmän käsittelyn jälkeen siirrytään uuden järjestelmän rakenteeseen ja tuodaan esille sen aiheuttamat muutokset sekä ratkaisut ongelmiin, joita vanhassa järjestelmässä oli.

Aluksi oli tarkoituksena myös kuvata uuden järjestelmän käyttöönottoa, mutta asioiden viivästymisen johdosta tämä vaihe jää alustavaksi. Uuden järjestelmän päätoimintaperiaatteet ovat kuitenkin hyvin selvillä ja ne kuvataan niin pitkälle, kuin tällä hetkellä on mahdollista.

Tampere Polytechnic University of Applied Sciences
Information Technology
Software Engineering

Peräjoki, Tommi

Delivery and install system for Oscar for Windows Enterprise
Resource Planning system.

Engineering Thesis

30 pages + 23 appendices

Thesis Supervisor

Senior Lecturer Jari Mikkolainen

Commissioning Company

Oscar Software Oy, valvojana M.Sc Jukka Harkki

May 2007

Keywords

ERP, install, delivery

ABSTRACT

This thesis describes the working process involved in developing a new delivery and setup system for Oscar for Windows ERP (Enterprise Resource Planning) developed and sold by Oscar Software Ltd. We also go through the stages of the current delivery and setup system.

At first we take a brief look at Oscar Software Ltd's background and describe the current delivery and setup system. Finally we move on to describe the new delivery and setup system and what changes and solutions it offers to the problems of the current system.

Introducing the new system for use was also supposed to be a part of this document, but delays in developing process caused this section to come out as a preliminary report. The basic principles of the new system are though well known and they are described in this document as thoroughly as possible at that current moment.

ALKUSANAT

Tämä työ on tehty Tampereen ammattikorkeakoulun opinnäytetyönä, toimeksiantajana Oscar Software Oy.

Kiitokset Oscar Software Oy:lle mahdollisuudesta toteuttaa tämä tutkintotyö. Erityisesti haluan kiittää Jukka Harkkia, tuotepäällikkö Pekka Mäkeä ja ohjelmistosuunnittelija Ari Leppälampea. Kiitän myös muita työssä auttaneita.

Tampereella 14. huhtikuuta 2007

Tommi Peräjoki

SANASTO

Schema	Schema on kokoelma tietokantaobjekteja kuten tauluja, indeksejä ja näyttöjä. Scheman omistaa tietokannan käyttäjä jolla on sama nimi kuin schemalla./1/
Client	työasema
Server	palvelin
OfW	Oscar for Windows
Master	esimerkiksi puhuttaessa Master-asennuksesta tarkoitetaan pääasennusta
ERP	Enterprise Resource Planning, toiminnanohjausjärjestelmä
UML	Unified Modeling Language, graafinen mallinnuskieli
ID	Identifier, yksilöivä tunniste
Session	istunto, yhteystapahtuma
Browser	selain
Tabpage	tab-sivu, välilehti

SISÄLLYSLUETTELO

TIIVISTELMÄ.....	i
ABSTRACT	ii
ALKUSANAT.....	iii
SANASTO.....	iv
1 JOHDANTO.....	1
2 NYKYISEN TOIMITUS- JA ASENNUSJÄRJESTELMÄN KUVAUS.....	2
2.1 Yleistä toimitus- ja asennusjärjestelmästä.....	2
2.2 Yleiskuvaus	2
2.3 Lotus Notes.....	3
2.4 OPak	3
2.5 Powergen	3
2.6 Lime.....	4
2.7 Oscar for Windowsin version asennus nykyisellä asennusjärjestelmällä.....	4
2.8 Nykyisen toimitus- ja asennusjärjestelmän ongelmat	5
3 UUDEN TOIMITUS- JA ASENNUSJÄRJESTELMÄN KUVAUS	7
3.1 Uuden toimitus- ja asennusjärjestelmän tuomat korjaukset ja ominaisuudet.....	7
3.2 Uuden järjestelmän määrittely ja toteutus	7
3.3 Uuden järjestelmän toteuttamisessa käytetyt työkalut.....	8
3.3.1 PowerBuilder-kehityympäristö	8
3.3.2 Oracle 10g -tietokantajärjestelmä.....	12
3.3.3 TOAD for Oracle -työkalu	14
3.3.4 NSIS-järjestelmä (NullSoft Scriptable Install System).....	15
3.4 Työn eteneminen	17
3.5 Uuden toimitus- ja asennusjärjestelmän työvaiheet	18
3.6 Uuden järjestelmän tietokannan tekeminen.....	18
3.7 Uuden tietokannan rakenne	18
3.8 Opakin ja LiMen yhdistäminen yhdeksi ohjelmaksi	20
3.8.1 Yleistä.....	20
3.8.2 OTO:n luokkahierarkia ja dynaamisuus	20
3.8.3 OTO:n ikkunat ja dataobjektit	22
3.8.4 Nykyisen järjestelmän tietokannan tietojen siirtäminen uuteen järjestelmään.....	23

3.9	Asennusjärjestelmän tekeminen	25
3.10	Toimitus- ja asennusprosessin vaiheet	25
3.10.1	Version tekeminen uudella järjestelmällä.....	25
3.10.2	Uuden asennusjärjestelmän palvelimen asennusohjelma	26
3.10.3	Uuden asennusjärjestelmän työaseman asennusohjelma.....	27
4	YHTEENVETO	29
	LÄHTEET	30

LIITTEET

- 1 Uuden toimitus- ja asennusjärjestelmän tietokannan UML-kuvaus
- 2 Opak-ohjelman kuvaus
- 3 Oscar for Windowsin version kääntäminen Opak-ohjelmalla
- 4 LiMe-ohjelman kuvaus ja käyttö
- 5 Palvelimen asennusohjelma
- 6 Työaseman asennusohjelma

1 JOHDANTO

Tämä työ tehtiin Oscar Software Oy:lle. Oscar Software on ohjelmistotalo, joka sijaitsee aivan Tampereen ydinkeskustassa, osoitteessa Kuninkaankatu 30 A. Yrityksessä työskentelee 10 henkilöä, joista kahdeksan Tampereen toimistolla, yksi Tuusulassa ja yksi Kotkassa. Yritys osti aikanaan Solteq Oy:ltä oikeudet Oscar for Windows -ohjelmaan, jonka tuotekehitystä se nyt jatkaa ja myy ohjelmaa uusille asiakkaille. Oscar for Windows (OfW) on ”Kappaletavarateollisuuden sekä teknisen tukkukaupan toiminnan- ja taloudenohjausjärjestelmä.”, ja käsittää tarkemmin eroteltuna seuraavat osa-alueet /2/

- Asiakkaat
- Varaston hallinta
- Tuotannon ohjaus
- Logistiikka
- Liikelaskenta
- Tiedonkeruu
- Liittymät
- Laskutus

Järjestelmän valtteja ovat muun muassa muokattavuus asiakkaan tarpeiden mukaiseksi ja usein päivittyvä ohjelman versio. Usein päivittyvä versio tarkoittaa, että asiakkaalle on mahdollista toimittaa tarvittaessa uusi versio ohjelmasta hyvinkin pienellä viiveellä. Uusi versio sisältää mahdollisten ohjelmasta löytyneiden virheiden korjauksia ja uusia ominaisuuksia, joita ohjelmaan on toivottu saatavaksi. Ohjelmiston ydin on Powerbuilder-kehitysympäristöllä ohjelmoitu ohjelma, joka käyttää Oraclen tietokantaympäristöä datan siirtoon, käsittelyyn ja säilöntään.

Tärkeänä osana järjestelmää toimii toimitus- ja asennusjärjestelmä, johon tässä työssä tarkemmin keskitytään.

2 NYKYISEN TOIMITUS- JA ASENNUSJÄRJESTELMÄN KUVAUS

2.1 Yleistä toimitus- ja asennusjärjestelmästä

Toimitus- ja asennusjärjestelmä voidaan jakaa kahteen osaan: toimitusjärjestelmään ja asennusjärjestelmään. Näiden järjestelmien avulla on tarkoituksena ylläpitää Oscar Softwaren asiakasrekisteriä ja lisenssejä sekä tuottaa asennusjärjestelmä, jolla Oscar Softwaren tuottama Oscar for Windows -ohjelma voidaan asiakkaalle asentaa. Toimitusjärjestelmän lopputuotoksena syntyy siis asennusjärjestelmä, joka on asiakkaittainen.

Toimitusjärjestelmän kautta voidaan suorittaa muun muassa seuraavia toimenpiteitä:

- Asiakasrekisterin ylläpito
- Lisenssien ylläpito
- Asiakkaittaisten tietojen ylläpito Oscar for Windows –ohjelmaan liittyvinä
- Oscar for Windows -ohjelman asennusjärjestelmän tuottaminen
- Historiatietojen avulla mahdollistuva seuranta siitä, mitä ja milloin on kenellekin asennettu

Asennusjärjestelmän kautta taas yksinkertaisesti suoritetaan Oscar for Windows -ohjelman asennus.

Seuraavaksi käydään ensin läpi kuvaus nykyisestä järjestelmästä ja sen toimintatavoista. Kuvauksen jälkeen pohditaan nykyisen järjestelmän ongelmia kappaleessa 2.4.

2.2 Yleiskuvaus

Nykyinen järjestelmä koostuu kahdesta eri ohjelmakokonaisuudesta, Opak ja Lime sekä niiden liitoksesta Lotus Notes -ohjelmaan. Uuden asiakkaan lisäämiseksi nykyiseen toimitus- ja asennusjärjestelmään on Lotus Notes-ohjelman kautta ensin

lisättävä asiakkaan tiedot Lotus Notesin kautta ylläpidettävään asiakastietokantaan, minkä jälkeen asiakkaalle on ollut mahdollista panna OfW:n versio kääntymään Opak-ohjelmalla. Kääntäminen tapahtuu Opakin sisältä ajettavalla Powergen - työkalulla. Tämän prosessin jälkeen siihen liitetään lisenssi LiMe-ohjelmalla.

2.3 Lotus Notes

Lotus on IBM:n omistama toimisto-ohjelmisto, johon kuuluvat muun muassa Lotus Notes ja Lotus Domino Server. Lotus Domino Server on palvelinohjelmisto, jonka tarjoamia palveluja Lotus Notes -ohjelmalla työasemalta käytetään. Lotus Notes Domino Serverin kautta tyypillisesti käytettyjä palveluja ovat muun muassa sähköposti-, kalenteri- ja erilaiset tietokannat, kuten asiakastietokanta, jota Oscar Softwaren nykyisessä toimitus- ja asennusjärjestelmässä myös hyödynnetään.

2.4 OPak

Opak on ohjelma, jolla käännetään asiakkaalle menevä OfW:n versio. Ohjelmaa ajetaan Oscar Softwaren sisäverkosta omalta tietokoneeltaan, jossa käyttöjärjestelmänä toimii Windows 2000 Advanced Server. Opakin muodostama OfW:n versio ei sinällään ole vielä siis ajettavissa, sillä siitä puuttuu asennusohjelma ja tarvittava lisenssi, joka yhdistää käännetyn OfW:n tietokantaan ja kertoo muita ohjelman käytön kannalta oleellisia tietoja.

Liitteessä 2 on vielä tarkempi kuvaus Opakista. Samalla käydään läpi kuinka Opakilla suoritetaan OfW:n version kääntäminen.

2.5 Powergen

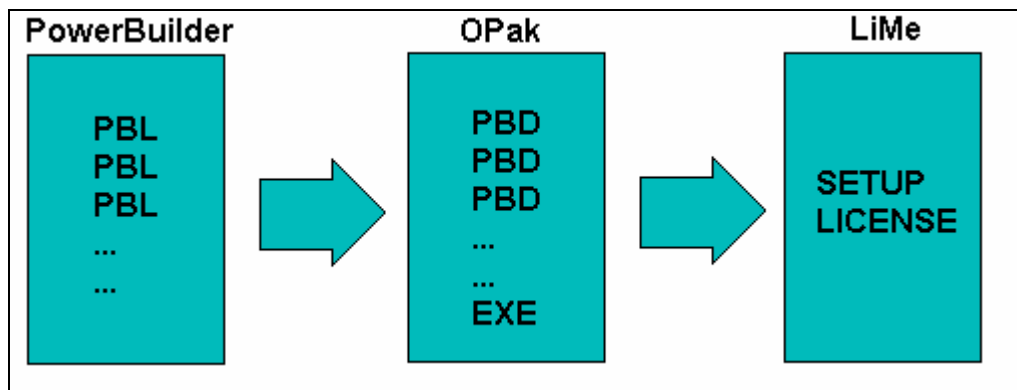
Powergen kääntää OfW:n lähdekoodin ajettavaan muotoon. OfW on ohjelmoitu PowerBuilder-työkalulla, jonka muodostama lähdekoodi sijaitsee PBL-tiedostoissa (PowerBuilder Library). Näistä PBL-tiedostoista Powergen-työkalu kääntää PBD-tiedostoja (PowerBuilder Dynamic Library) sekä EXE-tiedoston, joka käynnistää itse ohjelman.

2.6 Lime

Lime on ohjelmisto, jolla hallinnoidaan Lotus Notesin kautta lisättyjä asiakkaita ja luodaan heille lisensoijä käyttöoikeutta varten. Ohjelmalla liitetään Opakilla käännettyyn versioon Oscar for Windowsin lisenssi ja asennusohjelmistot ohjelman asentamiseksi.

Liitteessä 3 kerrotaan vielä tarkemmin LiMe-ohjelmasta ja samalla käydään läpi Oscar for Windows -ohjelman toimittaminen.

Kuvassa 1 on esitetty OfW:n version tekoprosessin vaiheet nykyisellä järjestelmällä.



Kuva 1 OfW:n version tekovaiheet

2.7 Oscar for Windowsin version asennus nykyisellä asennusjärjestelmällä

Oscar for Windowsin version asentamisessa on neljä vaihetta. Ensimmäisessä vaiheessa asennetaan ja luodaan Oraclen tietokanta, jota Oscar for Windows hyödyntää. Tämä toteutetaan Oraclen omalla asennusohjelmalla. Toisessa vaiheessa asennetaan Oscar for Windows uuteen tietokantaan. Kolmannessa vaiheessa ajetaan Oscar for Windowsin palvelimen asennusohjelma yhdellä verkon koneella. Tästä asennuksesta tulee ns. master-asennus, josta asennukset muille työasemille tehdään. Neljännessä vaiheessa työasemat asennetaan master-asennuksen pohjalta syntyneestä asennusohjelmasta.

Master-asennuksen avulla OfW:n päivittäminen on helppoa. Jokaisella työasemalla ei tarvitse erikseen käydä ajamassa päivitystä, vaan riittää, kun päivitetään master. Muut työasemat tarkastavat oman versionsa master-versiota vasten käynnistyessään ja lataavat tarvittaessa itselleen päivittyneen version masterilta. Seuraavassa on vielä tarkahko kuvaus sekä palvelimen että työaseman asennusohjelmasta.

Palvelimen asennusohjelma

Palvelimen asennusohjelma asentaa Oscar for Windowsin asennustiedostot palvelimelle, missä myös OfW:n Oracle-tietokanta sijaitsee, ja joko päivittää Oscarin tietokannan tai luo uuden tyhjän tietokannan. Itse Oscar for Windowsia ei vielä siis asenneta.

Työaseman asennusohjelma

Työaseman asennusohjelma sisältyy palvelimen asennusohjelmaan ja se asentuu palvelimen asennusohjelmalla määritettyyn paikkaan. Työaseman asennusohjelmalla asennetaan lopulta itse Oscar for Windows.

Liitteessä 5 käydään läpi ohjelman asentaminen palvelimelle ja liitteessä 6 työasemalle.

2.8 Nykyisen toimitus- ja asennusjärjestelmän ongelmat

Tärkeimpiä syitä uuteen toimitus- ja asennusohjelmistoon siirtymiseksi oli nykyisen järjestelmän yhteensopimattomuus OfW:n hyödyntämän Oraclen tietokantajärjestelmän uuden version kanssa ja se, että nykyisen järjestelmän muokkaaminen olisi tullut työläämmäksi kuin uuden järjestelmän toteuttaminen. Nykyinen asennusjärjestelmä on toteutettu Delphillä, ja järjestelmän tekijä ei enää työskentele yrityksessä. Dokumentointiakaan järjestelmästä ei ole. Nykyisen järjestelmän kehittäminen eteenpäin muodostui aivan liian hankalaksi.

Asennusohjelmat, joita nykyisessä järjestelmässä käytetään, sisältävät yhteensopimattomuun vuoksi turhia vaiheita. Esimerkiksi tietokannan päivitys, joka palvelimen asennuksessa aiemmin on ollut mahdollista tehdä, jätetään nyt tekemättä. Myös yhdessä työaseman asennuksen vaiheessa viitataan tiedostoon,

jolla ei nykyään ole enää mitään merkitystä. Nyt, jotta asennus etenee, pitää tämä tiedosto ensin luoda tyhjänä johonkin hakemistoon ja määrittää sen sijainti asennusohjelmassa.

Palvelimen ja työaseman asennuksessa käytetään järjestään aina samoja asennusohjelmia versiosta riippumatta, jolloin kaikki mahdolliset asiakkaittaiset erikoistoiminnot on täytynyt aina suorittaa käsin asennuksen jälkeen. Esimerkiksi jollakin asiakkaalla on voinut olla taulussa erikoisindeksi tai -rajoite.

Nykyinen järjestelmä koostuu monesta erillisestä komponentista. Opak-ohjelmalla käännetään OfW:n versio, LiMe-ohjelmalla suoritetaan lisensointi ja Lotus Notes -ohjelmalla ylläpidetään asiakasrekisteriä.

Opak-ohjelma sinänsä toimii ihan hyvin, mutta uuden järjestelmän kannalta se tulee turhaksi. Opak-ohjelman kautta suoritettava PowerGen-ajo on mahdollista sisällyttää uuteen järjestelmään ilman, että siitä pitäisi tehdä kokonaan erillistä ohjelmaa.

LiMe-ohjelmassa on turhiksi jääneitä ominaisuuksia, kuten Tuotteet-välilehti, josta oli mahdollista valita eri tuotteita kuin Oscar for Windows toimitettavaksi. Tätä ominaisuutta ei enää käytetty. LiMe vaatii käynnistyäkseen myös tunnuksen Lotus Notes -ohjelmaan asiakasrekisterin tietoja varten. Tällaisista liitoksista muihin ulkoisiin järjestelmiin haluttiin myös luopua.

Kaiken kaikkiaan nykyinen toimitus- ja asennusjärjestelmä on niin hajallaan ja jäänyt myös pikkuhiljaa kehityksestä jälkeen, että uuteen järjestelmään siirtymistä pidettiin tässä vaiheessaärkevänä vaihtoehtona. Uuden toimitus- ja asennusjärjestelmän tuomista korjauksista ja ominaisuuksista kerrotaan kappaleessa 4.1.

3 UUDEN TOIMITUS- JA ASENNUSJÄRJESTELMÄN KUVAUS

3.1 Uuden toimitus- ja asennusjärjestelmän tuomat korjaukset ja ominaisuudet

Uusi järjestelmä korvaa käytössä olevan järjestelmän ja sen komponentit. Nykyisen järjestelmän erilliset ohjelmat integroidaan yhdeksi entistä selkeämmäksi kokonaisuudeksi, joka ei ole riippuvainen mistään muusta toisesta järjestelmästä samaan tapaan, kuin esimerkiksi LiMe ja Opak ovat nykyisessä järjestelmässä riippuvaisia Lotus Notesin kautta ylläpidettävästä asiakasrekisteristä. Myös nykyisessä järjestelmässä olevat yhteensopivuusongelmat häviävät uuden järjestelmän myötä. Yhteensopivuusongelmiahan ilmeni asennusohjelmistojen ja tietokantayhteyden kanssa. Uusi järjestelmä mahdollistaa muun muassa seuraavia asioita:

- toimitus- ja asennusjärjestelmä saadaan koodipohjaltaan entistä hallittavammaksi
- asennusohjelmat saadaan tehdyksi yleisestikin käytössä olevalla työkalulla (NSIS)
- lisensointiin saadaan lisäominaisuuksia, muun muassa moduulittainen lisensointi
- demolisensseihin saadaan aikarajat
- järjestelmää on mahdollista käyttää etäjärjestelmän kautta
- asennus tulee entistä helpommaksi ja mahdollistaa entistä laajemmin asiakkaiden itse tekemät asennukset

3.2 Uuden järjestelmän määrittely ja toteutus

Uuden järjestelmän määrittelyn teki Oscar Softwaren työntekijä Jukka Harkki. Minun tehtävänäni oli toteuttaa järjestelmä määrittelyn mukaiseksi. Määrittely on alustava ja toteutuksen edetessä sitä muutetaan tarvittaessa. Järjestelmälle annettiin alustavasti nimeksi OTO. Vastedes tässä dokumentissa uutta järjestelmää käsitellään myös nimellä OTO.

3.3 Uuden järjestelmän toteuttamisessa käytetyt työkalut

Uuden järjestelmän toteutuksessa hyödynnetään seuraavia työkaluja:

- Powerbuilder, versio 10.2.1
- Oracle, versio 10g (10.2.0.1)
- Toad, versio 9.0.1.8
- NullSoft Scriptable Install System (NSIS), versio 2.24

Seuraavissa kappaleissa esitellään jokainen työkalu lyhyesti.

3.3.1 PowerBuilder-kehitysympäristö

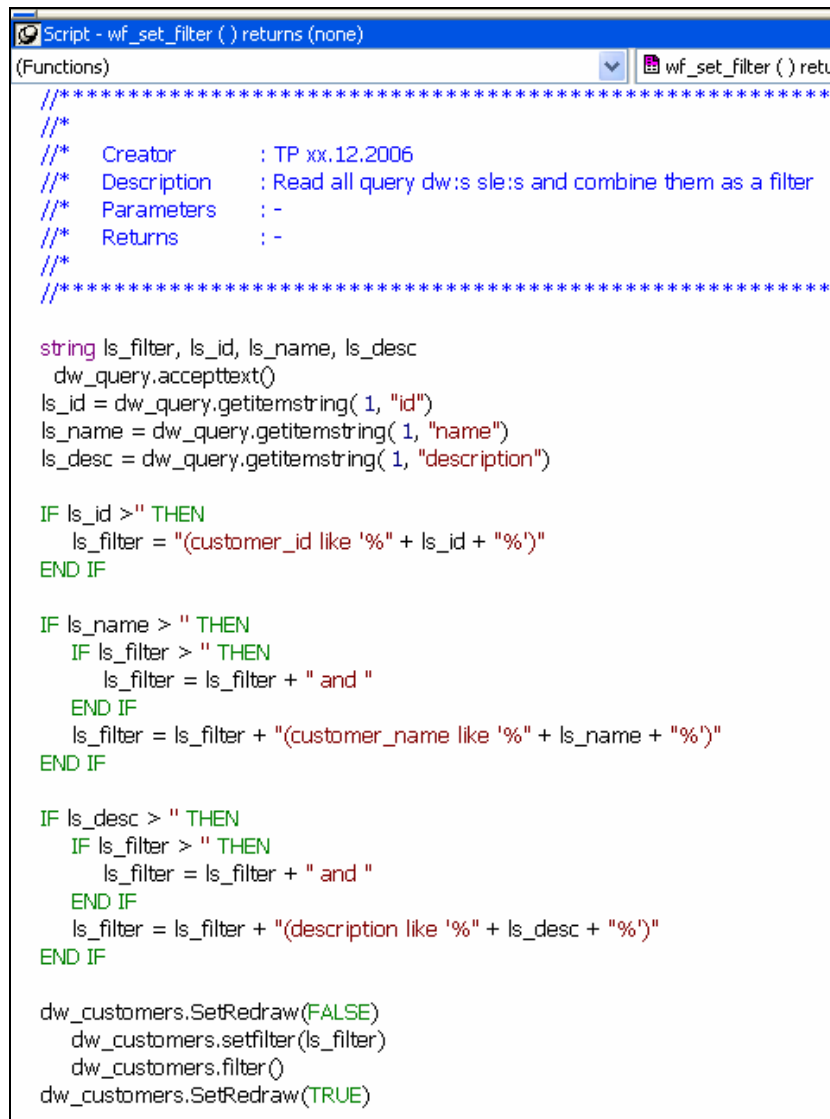
OTO ohjelmoitiin PowerBuilder-kehitysympäristön versiolla 10.2.1. PowerBuilder on Sybasen kehitysympäristö, jolla voidaan toteuttaa ohjelmistoja eri tietokantayhteyksien päälle. Ohjelmalla tuotetaan tyypillisesti client-server-mallin mukaisia sovelluksia ja sitä voisi verrata esimerkiksi Microsoftin Visual Studio -kehitysympäristöön. Tällaisissa ympäristöissä objektien muoto määritetään hiirellä piirtämällä ja ohjelman logiikka toteutetaan taustalle. Ohjelman tuottama koodi on DataWindow-kontrollia (tarkemmin jäljempänä) lukuun ottamatta aitoja Windows-komponentteja ja niihin sidottua viestiliikennettä.

Käydään läpi muutama OTO:on läheisesti liittyvä PowerBuilder peruskäsite:

PowerScript

PowerScript on PowerBuilderin ohjelmointikieli, jolla PowerBuilderin ohjelmat ohjelmoidaan. Se tukee olio-ohjelmointia, mutta olio-ohjelmoinnin logiikkaa ei silti ole pakko käyttää. OTO toteutetaan olio-ohjelmoinnin periaatteina noudattaen ja toteuttamisessa käytetään mahdollisimman paljon jo valmiina olevia rakenteita PFC-kirjastoista (PowerBuilder Foundation Classes). Niistä periytetään omat tarvittavat objektit ohjelman rakentamista varten. Kuvassa 2 on näyte PowerScriptin syntaksista. Käytössä on olio-ohjelmoinnista tuttuja ohjelmointitapoja ja normaalit rakenteet, kuten IF-ELSE, DO-LOOP, CASE

toimivat myös. Ohjelmakoodin rivejä ei tosin ohjelmoinnissa jo yleensä totuttuun tapaan päätetä PowerBuilderissa puolipisteellä.



```
Script - wf_set_filter ( ) returns (none)
(Functions) wf_set_filter ( ) retu

/**
/**
/** Creator      : TP xx.12.2006
/** Description  : Read all query dw:s sle:s and combine them as a filter
/** Parameters   : -
/** Returns     : -
/**
/**
/**
string ls_filter, ls_id, ls_name, ls_desc
dw_query.accepttext()
ls_id = dw_query.getitemstring( 1, "id")
ls_name = dw_query.getitemstring( 1, "name")
ls_desc = dw_query.getitemstring( 1, "description")

IF ls_id > " THEN
    ls_filter = "(customer_id like '%" + ls_id + "%)"
END IF

IF ls_name > " THEN
    IF ls_filter > " THEN
        ls_filter = ls_filter + " and "
    END IF
    ls_filter = ls_filter + "(customer_name like '%" + ls_name + "%)"
END IF

IF ls_desc > " THEN
    IF ls_filter > " THEN
        ls_filter = ls_filter + " and "
    END IF
    ls_filter = ls_filter + "(description like '%" + ls_desc + "%)"
END IF

dw_customers.SetRedraw(FALSE)
dw_customers.setfilter(ls_filter)
dw_customers.filter()
dw_customers.SetRedraw(TRUE)
```

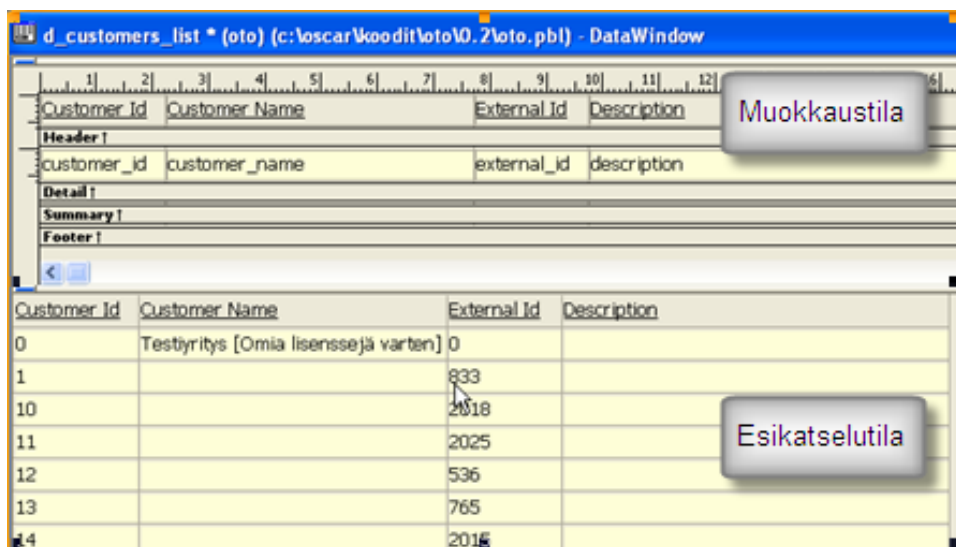
Kuva 2 Näyte PowerScript-ohjelmointikielen syntaksista

DataWindow

Datawindow, dataikkuna, on datankäsittelyobjekti jota voidaan käyttää muokkaamaan, näyttämään ja lisäämään määritettyyn tietokantaan dataa. Se edustaa sen kautta määritettyä dataobjektia, joka taas osoittaa johonkin tietokannan tauluun./3/

DataObject

Dataobject, dataobjekti, on komponentti jolla voidaan käsitellä kannan tietoja tai sen kautta voidaan kerätä tietoja käyttäjältä. Objekti liitetään aina edellä mainittuun DataWindow-objektiin, joka sisältää komennot, joilla dataobjektia käsitellään. /3/ Tyypillisimmät OTO:ssa käytetyt dataobjektit ovat grid- tai freeform-tyyppisiä. Grid-tyyppisessä dataobjektissa tiedot näytetään ruudukossa. Kuvassa 3 on esitetty asiakaslistauksen suorittava grid-tyyppinen dataobjekti PowerBuilderin muokkaustilassa. Ruudun yläosassa on muokkaustila, alaosassa on ruutu esikatselutilassa.



Kuva 3 Grid-tyyppinen PowerBuilder-dataobjekti

Freeform-tyyppisessä dataobjektissa näyttöä voi vapaasti muokata. Kuvassa 4 on esitetty asiakkaan ylläpidon suorittava freeform-tyyppinen dataobjekti PowerBuilderin muokkaustilassa.

Mikäli dataobjekti on liitetty johonkin kantaan, sille voidaan määrittää hakuetoja, joilla alustavasti kannasta haettuja tietoja voidaan rajata.

Tab

Tab-objekti on kontrolli, joka toimii liitäntäalustana tabpageille, välilehdille.

TabPage

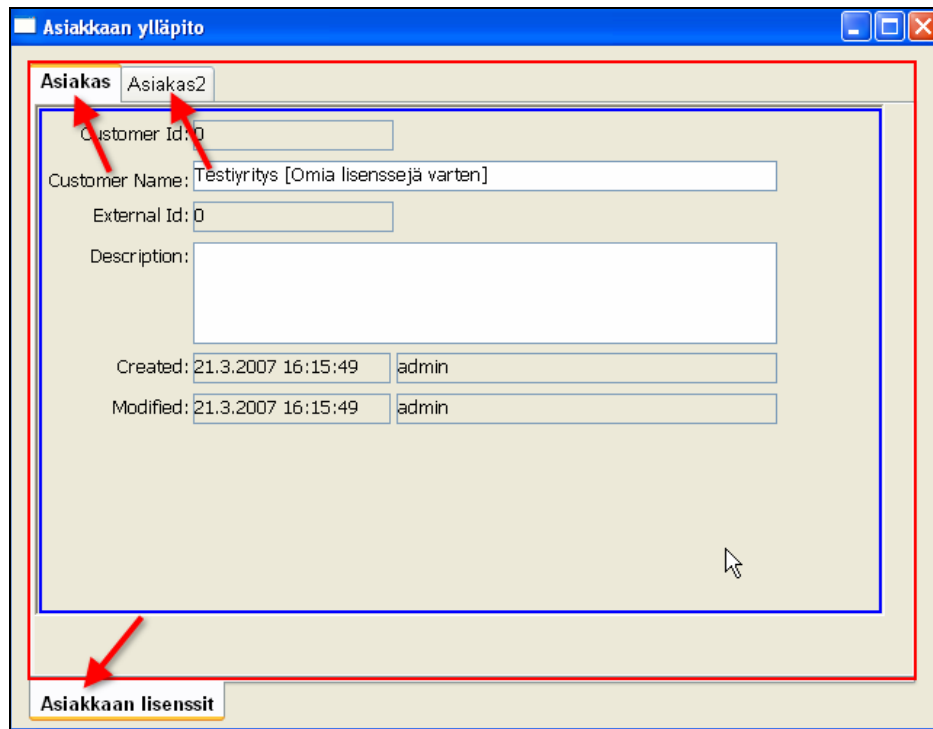
TabPage, välilehti, liitetään tab-objektiin, jotka yhdessä muodostavat yhtenäisen kokonaisuuden. Välilehtiä voi olla yhdellä tabilla monia. Kuvassa 5 on esitetty tab- ja tabpage-objektien muodostama kokonaisuus, johon on liitetty dataikkunan kautta dataobjekti. Kuvassa tab-objekti on taustalla rajattuna punaisella. Välilehdet on merkitty kuvaan punaisilla nuolilla. Dataobjekti on kuvassa rajattuna sinisellä.

The image shows a screenshot of a PowerBuilder data object design. It features a 'Header' section with fields for Customer Id, Customer Name, External Id, and Description, and a 'Detail' section with corresponding data fields. A mouse cursor is visible over the Description field in the header section.

Header		
Customer Id:	customer_id	
Customer Name:	customer_name	
External Id:	external_id	
Description:	description	
Created:	created	created_by
Modified:	modified	modified_by

Detail		
Customer Id:	1	
Customer Name:		
External Id:	833	
Description:		
Created:	21.3.2007 16:15:49	admin
Modified:	21.3.2007 16:15:49	admin

Kuva 4 Freeform-tyyppinen PowerBuilder-dataobjekti



Kuva 5 Tab, tabpage ja dataobjekti yhdistettynä

Window

Window-objekti on pohja, johon ikkunat rakennetaan. Tälle pohjalle voidaan liittää myös painikkeita ja muita objekteja ohjaamaan ohjelman toimintaa. Kuvassa 5 esitetty ikkuna on window-objekti, johon on liitetty tab-objekti (rajattu punaisella), ja tähän tabiin on liitetty tab-sivut Asiakas, Asiakas2 ja Asiakkaan lisenssit. Asiakas-tab-sivun kautta on liitetty näkyväksi datawindow, jonka kautta näkyy dataobjekti, mistä nähdään asiakkaan ylläpitoon liittyvä ruutu (rajattu sinisellä).

3.3.2 Oracle 10g -tietokantajärjestelmä

Koska Oscar for Windows -ohjelma käyttää Oraclen tietokantaa, oli siitä jo kokemusta, ja muutenkaan ei ollut syytä vaihtaa jo opittua ja käytössä olevaa kantajärjestelmää miksikään muuksi, joten asennus- ja toimitusjärjestelmässä kantana käytettiin myös Oraclea ja siitä versiota 10g.


Oracle 10g -tietokantajärjestelmä on Oraclen kehittämä. Oraclea pidetään yleensä vaihtoehtona, kun käsitellään suuria määriä dataa. Sitä on mahdollista käyttää eri ympäristöissä, kuten Windows-, Linux- ja Unix-ympäristöissä./4/

Oraclen kantaa voidaan käsitellä Windowsissa eri tavoilla, joista seuraavassa on lueteltu muutama:

- SQL Plus -ohjelma, käytetään komentokehoteen kautta
- SQL Worksheet, Windows-ohjelma
- TOAD for Oracle, Windows-ohjelma, jolla kannan käsittely on huomattavasti havainnollisempaa kuin SQL Plus tai SQL Worksheet -ohjelman kautta. SQL Plus ja SQL Worksheet -ohjelmat ovat kuitenkin omiaan esimerkiksi valmiiden scripttien ja lyhyiden operaatioiden tekemiseen. TOAD:illa suurimmaksi osaksi kannan scriptejä kuitenkin ajettiin ja muokattiin ja dataa selailtiin.

Kaksi ensimmäisenä mainittua ohjelmaa tulevat Oraclen mukana, TOAD on maksullinen ja kolmannen osapuolen kehittämä ohjelma. TOAD:ista kerrotaan seuraavassa kappaleessa vielä tarkemmin.

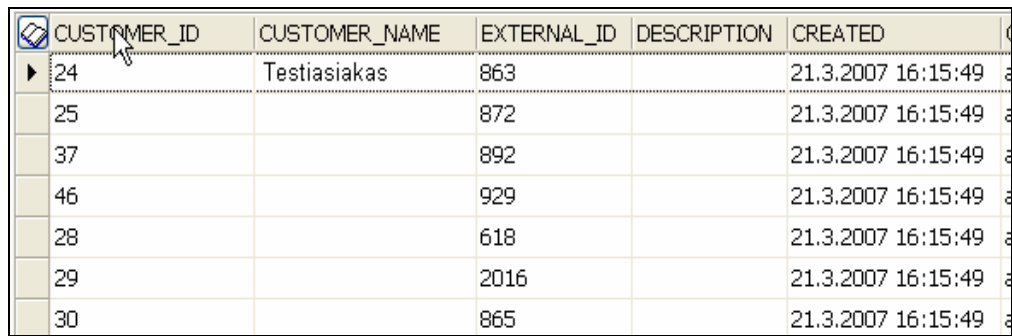
SQL Plus -ohjelmalla suoritettavat kyselyt paisuvat helposti lukemattomiksi, kuten kuvasta 6 voidaan todeta. Kuvassa 7 on sama data nähtävissä TOAD:illa.



```

c:\> Komentorivi - sqlplus oto/oto@tst
SQL> select * from oto_customers where customer_id='20';
CUSTOMER_ID
-----
CUSTOMER_NAME
-----
EXTERNAL_ID
-----
DESCRIPTION
-----
CREATED      CREATED_BY      MODIFIED
MODIFIED_BY
-----
20
CUSTOMER_ID
-----
CUSTOMER_NAME
-----
```

Kuva 6 SQL Plus-ohjelma



CUSTOMER_ID	CUSTOMER_NAME	EXTERNAL_ID	DESCRIPTION	CREATED
24	Testiasiakas	863		21.3.2007 16:15:49
25		872		21.3.2007 16:15:49
37		892		21.3.2007 16:15:49
46		929		21.3.2007 16:15:49
28		618		21.3.2007 16:15:49
29		2016		21.3.2007 16:15:49
30		865		21.3.2007 16:15:49

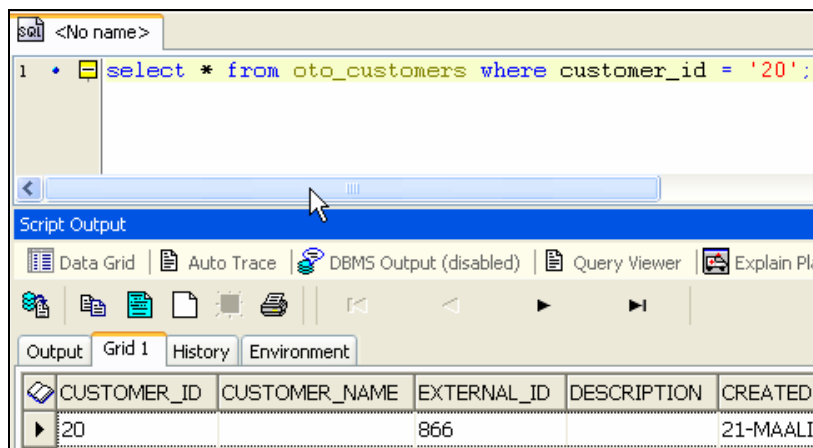
Kuva 7 TOAD-ohjelma

3.3.3 TOAD for Oracle -työkalu

TOAD for Oracle -työkalulla Oraclen tietokannan käsittely ja datan selailu on huomattavasti helpompaa kuin käsin komennettavilla SQL Plus ja SQLWorksheet -työkaluilla./5/

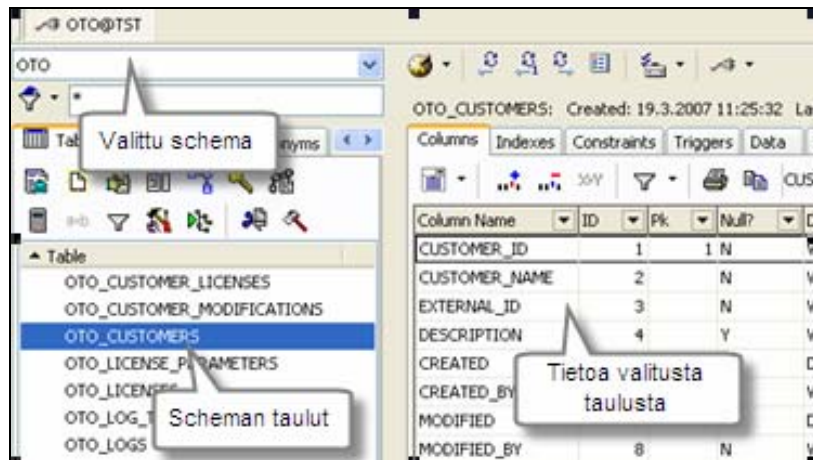
TOAD:in perustyökaluihin kuuluvat:

- Editori (kuva 8), jolla voidaan tehdä samat operaatiot kuin SQL Plussalla ja SQL Worksheetilla, mutta kyselyjen kautta saatua dataa voidaan käsitellä lukemattomilla eri tavoilla, esimerkiksi muuttaa suoraan erilaisiksi formaateiksi, kuten Microsoftin Excel-tiedostoksi tai tehdä datasta suoraan Insert-lauseita.



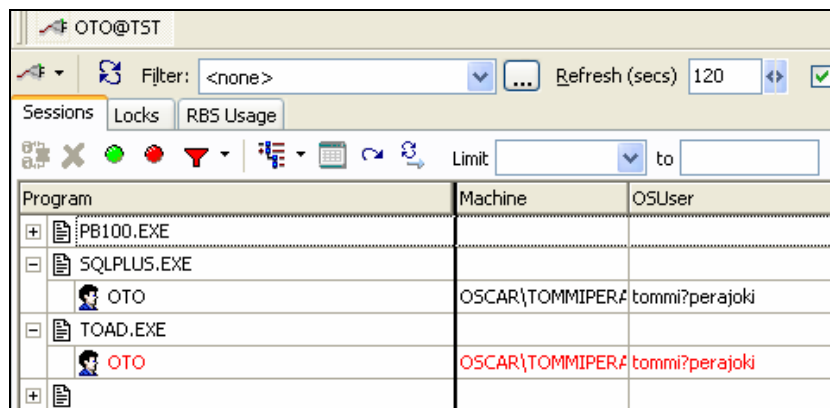
Kuva 8 TOAD ja sen sisäinen editor

- Schema Browser, jolla Oraclen Schemoja eli kantoja voidaan kätevästi selailla (kuva 9)



Kuva 9 TOAD ja Schema Browser

- Session Browser, jolla voi tarkkailla tietyn kannan tai kantojen tilannetta (kuva 10)



Kuva 10 TOAD ja Session Browser

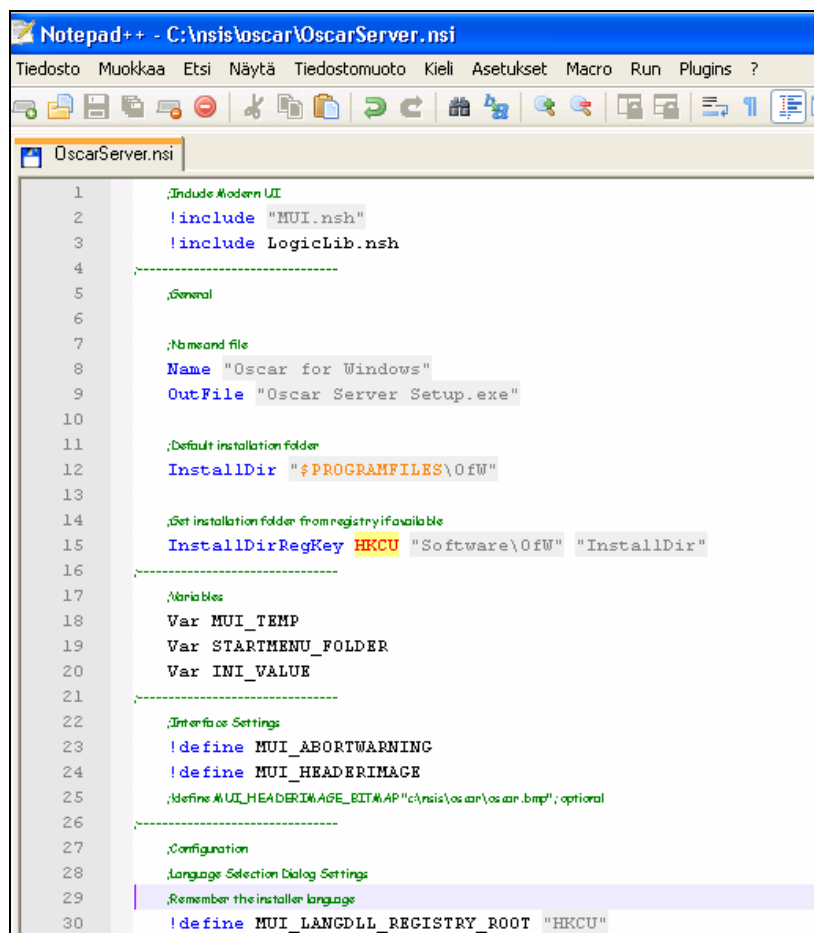
3.3.4 NSIS-järjestelmä (NullSoft Scriptable Install System)

NullSoft Scriptable Install System on OpenSource ideologiaan perustuva järjestelmä, jonka on kehittänyt NullSoft. NullSoft on tunnettu myös kehittämästään Winamp-mediasoittimesta ja Shoutcast-nettiradiojärjestelmästä. NSIS-järjestelmällä voidaan toteuttaa Windowsissa toimivia asennusohjelmia. /6/

Asennusohjelman toteuttaminen tehdään scriptejä hyödyntäen. Scriptit ohjelmoidaan käsin tekstieditorilla. Scriptien toteuttamiseksi löytyy myös työkaluja ja tekstieditoreita, jotka tukevat NSIS-järjestelmän syntaksia. Toteutetut scriptit

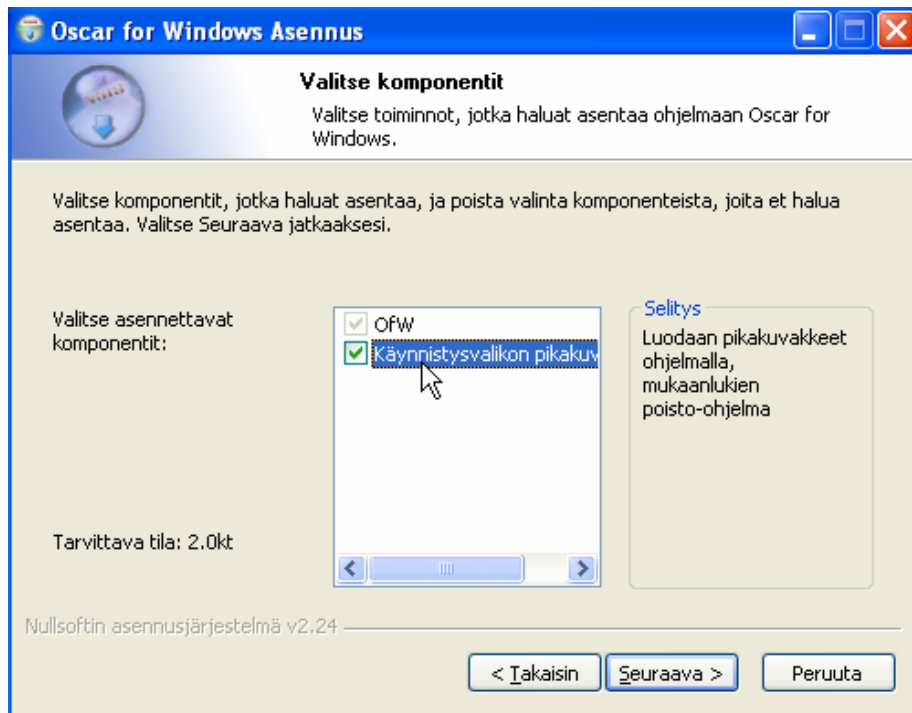
voidaan kääntää joko Windowsin kautta MakeNSISW-ohjelmalla tai komentokehotteesta MakeNSIS-ohjelmalla. Kuvassa 11 esitellään, miltä NSIS-scriptin syntaksi näyttää Notepad++-tekstieditorin läpi editoituna.

NSIS-järjestelmällä on jo toteutettu perusrunko testausmielessä Oscar for Windows tiedostojen asentamiseksi. Kuvassa 12 on esitetty ruutu, jossa käyttäjä voi valita asennettavat komponentit.



```
1      ;Include Modern UI
2      !include "MUI.nsh"
3      !include LogicLib.nsh
4
5      ;General
6
7      ;Name and file
8      Name "Oscar for Windows"
9      OutFile "Oscar Server Setup.exe"
10
11     ;Default installation folder
12     InstallDir "$PROGRAMFILES\OfW"
13
14     ;Set installation folder from registry if available
15     InstallDirRegKey HKCU "Software\OfW" "InstallDir"
16
17     ;Variables
18     Var MUI_TEMP
19     Var STARTMENU_FOLDER
20     Var INI_VALUE
21
22     ;Interface Settings
23     !define MUI_ABORTWARNING
24     !define MUI_HEADERIMAGE
25     ;define MUI_HEADERIMAGE_BITMAP "$NSIS\oscar\oscar.bmp"; optional
26
27     ;Configuration
28     ;Language Selection Dialog Settings
29     ;Remember the installer language
30     !define MUI_LANGDLL_REGISTRY_ROOT "HKCU"
```

Kuva 11 NSIS-scriptin syntaksia



Kuva 12 Uuden asennusohjelman testiversion asennettavien komponenttien valitsemisruutu

3.4 Työn eteneminen

Määrittely oli alustavasti valmis jo syksyllä 2006, mutta johtuen kiireellisemmistä töistä työn toteuttaminen jäi vähemmälle. Yleensäkin työn toteutus eteni vaiheittain johtuen osaltaan siitä, että työn laajuutta ei ehkä täysin ymmärretty. Verkkaista tahtia ylläpiti myös se asia, että uusiin järjestelmiin piti ensin tutustua jotta niitä voitiin kunnolla käyttää. Tässä viitataan juurikin TOAD-ja NSIS-ohjelmiin sekä osittain myös Powerbuilderiin. Tein toteutusta pääosin yksin. Työ eteni lähinnä silloin, kun muilta kiireellisemmiltä asioilta ehti.

Jukka Harkki toimi työn valvojana suorittaen välillä työn sisällön tarkastelua, ja toteutti myös järjestelmän ytimeen logiikan, mikä liittyi ohjelman näyttöjen dynaamisuuteen. Häneltä sain myös ohjeita kun työ ei edennyt, sekä tarkennuksia siihen miten missäkin vaiheessa ohjelman tulee toimia. Apua sain myös tuotepäällikkö Pekka Mäeltä sekä sovellussuunnittelija Ari Leppälammelta.

Aluksi järjestelmässä oli tarkoituksena toteuttaa asennusohjelmat WiX -työkalua käyttäen (Windows Installer XML), mutta se vaihdettiin NSIS-työkaluun sen

selkeämmän rakenteen takia. Molemmat työkalut ovat Open Source pohjaisia eli ilmaisia käyttää.

3.5 Uuden toimitus- ja asennusjärjestelmän työvaiheet

Uuden järjestelmän toteuttamisessa on monia eri vaiheita. Mitään suoranaista järjestystä näillä ei ole, vaan niitä tehdään osittain päällekkäin. Jokaista vaihetta on mahdollista tehdä erikseen toisistaan riippumatta, mutta kokonaisuuden testaamiseksi kaikkien pitää lopulta toimia.

Tehtävälisellä ovat seuraavat työvaiheet:

- Pehdytään nykyisen asennus- ja toimitusjärjestelmän logiikkaan
- Tehdään kanta uudelle asennus- ja toimitusjärjestelmälle
- Siirretään vanhat asiakastiedot uuteen järjestelmään
- Tehdään asennusohjelmistot server- ja client-asennusta varten
- Kootaan LiMe sekä Opak yhdeksi Powerbuilderilla tehdyksi ohjelmaksi
- Testataan ohjelmaa

3.6 Uuden järjestelmän tietokannan tekeminen

Uutta tietokantaa varten tehtiin oma SQL-scriptinsä, jolla kanta luotiin. Kantaan syötettiin aluksi testidataa ohjelman kehittämistä varten. Nykyisen kannan tietoja ei heti siirretty uuteen kantaan, koska se ei ollut välttämätöntä tässä vaiheessa.

3.7 Uuden tietokannan rakenne

Liitteessä 1 on UML-kuvaus tietokannan rakenteesta.

Uusi tietokanta toimii seuraavasti:

- Jokaisessa taulussa on tiedot milloin tietoa on tauluun lisätty ja milloin sitä on muokattu

- Customers-tilussa on asiakkaan ID ja nimi sekä kuvaus asiakkaasta. Tilussa on myös määriteltyä asiakasnumero, jolla määritetään hakemiston ID, jossa asiakkaan tietoja säilötään.
- Customer Licenses -tilussa liitetään asiakkaaseen lisenssi, joka sijaitsee Licenses-tilussa. Yhteensä asiakkaaseen voi liittää monta lisenssiä.
- Licenses-tilussa on määriteltyä lisenssin ID, nimi, lisenssien lukumäärä ja aikaväli milloin lisenssi on voimassa
- License Parameters -tilussa on lueteltuna parametrit, jotka liittyvät tiettyyn lisenssiin. Jokaiselle parametrille on annettu arvo ja parametrilla on viittaus parametri tauluun, joka kertoo tarkemmin parametrin tarkoituksesta. Yhteensä lisenssiin voi määrittää vain kerran saman parametrin.
- Parameters-tilussa on tieto siitä, mitä parametri tekee. Siellä myös määritetään onko jokin parametri automaattisesti olemassa lisenssillä, kun lisenssi luodaan, sekä onko parametrin olemassaolo pakollinen lisenssin oikeamuotoisuuden säilymiseksi. Lisäksi määritetään oletusarvo parametrille
- Customer Modifications -tilussa määritetään asiakkaaseen liittyvät modifikaatiot eli erikoiskirjastot, -tiedostot ja -scriptit
- Modifications-tilussa määritetään eri modifikaatiot. Modifikaatiosta voidaan määrittää onko se yleinen modifikaatio. Tauluun määritetään myös modifikaation sijainti. Modifikaatioita on erityyppisiä. Nämä tyypit on määritelty Modification Type -luettelona. Tyyppejä ovat alustavasti:
 - PBL-LIBRARY : erikoiskirjasto
 - PBL-RUNTIME : DLL-tiedosto PowerBuilder-ohjelman ajamiseksi
 - SQL-SCRIPT : tietokantaan ajettava erikoisscripti
 - SYS-SCRIPT : SYS-käyttäjänä tietokantaan ajettava erikoisscripti
- Version Builds -tilussa säilytetään tieto siitä, mille asiakkaalle versio on tehty, kuka sen teki, mikä on version Build-numero ja version ID-numero sekä mihin lokiin version tiedot tallennetaan LogId:tä hyödyntäen
- Users-tilussa ylläpidetään OTO:n käyttäjien tietoja.
- Logs-tilussa säilytetään tietoja eri versioiden ajoista. Lokitiedolla voi olla eri statuksia riippuen toiminnosta mihin se liittyy. Vaihteita ovat:
 - CREATED : loki luotiin
 - STARTED : lokiajo alkoi

- FINISHED lokiajo loppui
- FAILED : lokiajo epäonnistui
- Log Texts -taulu sisältää rivit jotka yhdistetään tiettyyn lokiin LogId:n avulla
- Registry-aulussa ylläpidetään OTO:n suorittamisen kannalta tärkeitä tietoja, kuten esimerkiksi version tekemisessä tarvittavien hakemistojen sijaintia

3.8 Opakin ja LiMen yhdistäminen yhdeksi ohjelmaksi

3.8.1 Yleistä

OPakin ja LiMen logiikka rakennetaan uudelleen alusta yhdeksi kokonaisuudeksi, OTO:ksi. Uuden järjestelmän ajamiseksi pitää koneelta siis löytyä Oraclen Client-asennus ja siihen määriteltynä tarvittavat tiedot yhteyden avaamiseksi OTO:n tietokantaan.

Vasta kannan tekemisen jälkeen oli järkevää aloittaa uuden järjestelmän tekeminen PowerBuilderilla, jotta oli dataa mitä ohjelmaa tehtäessä oli mahdollista hyödyntää. Ilman kantaa ei PowerBuilderilla olisi voinut alkaa järkevästi ohjelmaa edes tekemään, koska oleellisin osa kehitysympäristössä suoritetuista prosesseista nojautuivat siihen.

3.8.2 OTO:n luokkahierarkia ja dynaamisuus

OTO:n pohjana toimii dynaaminen ohjelman rakentaminen. Halutut näytöt ladataan INI-tiedoston kautta. INI-tiedosto sisältää tiedon muunmuassa siitä, mitä tabpageja ladataan millekin tabille ja kuinka paljon. Tällä saadaan aikaiseksi käyttäjäkohtaisten näyttöjen luominen ja hallittavampi ympäristö.

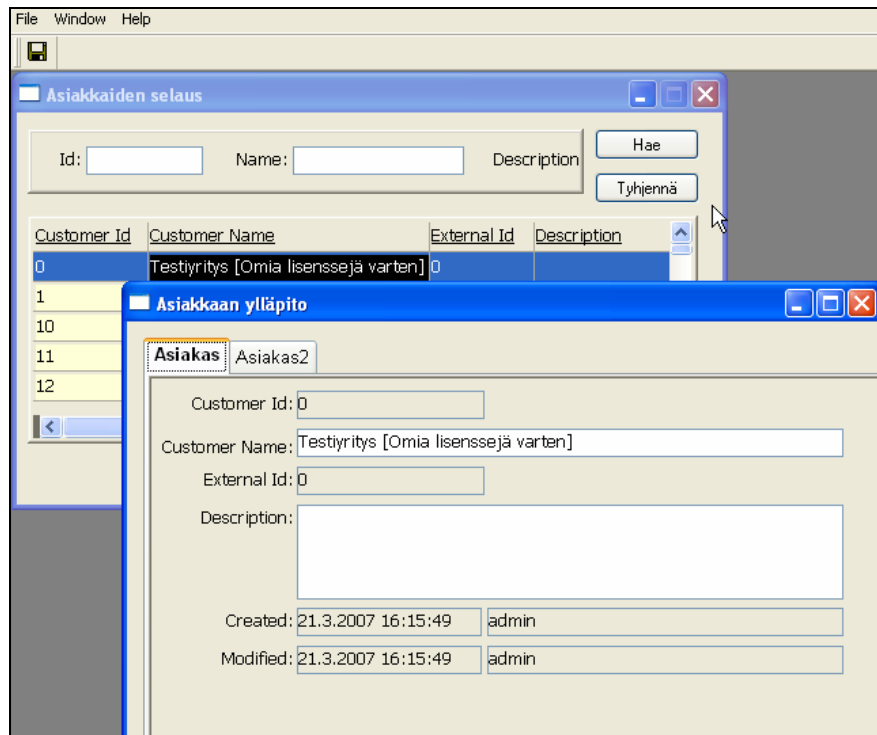
Ohjelmassa on olemassa kahta erityyppistä sheet-objektia, List (lista) ja Edit (ylläpito). List-tyyppiselle sheet-objektille toteutetaan tietokannan tietojen listaukseen tarvittavaa logiikkaa ja edit-tyyppiselle sheet-objektille toteutetaan tietokannan muokkaukseen tarvittavaa logiikkaa. Edelleen on olemassa list- ja edit-tyyppiset DataWindow-objektit. Näissä on myös toteutettuna tyyppistä riippuen oma logiikkansa. Eri tyyppisten ikkunoiden ja perinnän avulla on mahdollista

automatisoida suurin osa normaalista navigointi- ja logiikkakoodista, mikä lisää tuottavuutta ja pienentää virheherkkyyttä.

Ohjelmassa on yleisesti siis käytössä kahdentyyppisiä ikkunoita: listaikkunoita ja ylläpitoikkunoita. Näiden välillä on linkki toisiinsa, eli kun käyttäjä esimerkiksi asiakkaiden listausikkunassa hiirellä kaksoisnäpäyttää asiakasta, avautuu tälle asiakkaalle ylläpitoikkuna, jolla voidaan ylläpitää kyseisen asiakkaan tietoja (Kuva 13).

Asiakkaan ylläpitoikkunassa on lista asiakkaaseen liitetyistä lisensseistä, jolloin lisenssiä hiirellä kaksoisnäpäyttämällä saadaan näkyviin lisenssiin liittyvä ylläpitoikkuna. Näin edelleen lisenssin ylläpitoikkunassa voi taas olla listaus esimerkiksi lisenssiin liittyvistä parametreista ja niistä jotain hiirellä kaksoisnäpäyttämällä avautuu kyseiselle parametrille jälleen ylläpitoikkuna. Tämä logiikka toistuu samanlaisena kaikissa OTO:n näytöissä.

Tiedot tallentuvat painamalla kuvassa 13 vasemmassa yläruudussa näkyvää levykkeen kuvaa. Mikäli ikkuna suljetaan tallentamatta tietoja, ohjelma kysyy tallennetaanko tiedot ennen ikkunan sulkemista.



Kuva 13 Asiakkaiden selausikkuna ja sen kautta avattu asiakkaan ylläpitoikkuna

3.8.3 OTO:n ikkunat ja dataobjektit

OTO:ssa on tehtynä tällä hetkellä asiakkaiden ja lisenssien listaus- ja ylläpitoikkunat. Nämä ikkunat ovat niitä ikkunoita, joita ensimmäisenä ohjelman käytössä tullaan tarvitsemaan, joten niistä oli järkevää aloittaa ohjelman tekeminen. Kun näiden ikkunoiden logiikka on saatu toimimaan, on muut ikkunat helppo toteuttaa samaan tyyliin. OTO:ssa oli ideana toteuttaa mahdollisimman paljon ikkunoiden peruslogiikasta periyttäviin luokkiin, jotta sitä voitaisiin hyödyntää yleisesti kaikissa ikkunoissa. Tyyppejään OTO:ssa oli käytössä kaksi: listaus- ja ylläpito. Ikkunat ja dataobjektit tulevat lopulta olemaan ovat seuraavat:

Ikkunat

- Asiakkaiden selaus- ja ylläpitoikkunat
- Asiakkaan lisenssien selaus- ja ylläpitoikkunat
- Lisenssien selaus- ja ylläpitoikkunat
- Parametrien selaus- ja ylläpitoikkunat
- Käyttäjien selaus- ja ylläpitoikkunat
- Lokin selausikkunat

- Modifikaatioiden selaus- ja ylläpitoikkunat
- Versiointekoikkunat

Objektit

- Asiakkaiden selaus- ja ylläpito-objektit
- Asiakkaiden lisenssien selaus-objekti
- Asiakkaan modifikaatioiden selaus- ja ylläpito-objektit
- Lisenssien selaus- ja ylläpito-objektit
- Lisenssin parametrien ylläpito-objekti
- Parametrien selaus- ja ylläpito-objektit
- Version teko -ikkunaan tarvittavat objektit
- Rekisterin selaus- ja ylläpito-objektit
- Käyttäjien selaus- ja ylläpito-objektit
- Lokitietojen selaus- ja katseluobjektit

Näistä asiakkaisiin ja lisensseihin liittyvien ikkunoiden ja objektien tekeminen on vasta aloitettu.

Jäljellä olevista ikkunoista haasteellisin toteuttaa on Version teko -ikkuna. Ikkunan kautta tullaan käynnistämään lopullisen asiakkaalle menevän version teko.

Tämä ikkuna poikkeaa muista ikkunoista siinä, että sen kautta ei sinänsä ylläpidetä mitään tietoa. Ikkunan kautta kaikki muissa ikkunoissa ylläpidetyt tiedot liitetään toisiinsa ja näistä tiedoista saadaan aikaiseksi Oscar for Windowsin versio.

3.8.4 Nykyisen järjestelmän tietokannan tietojen siirtäminen uuteen järjestelmään

Tärkeänä vaiheena uuden järjestelmän toteuttamisessa on myös nykyisen järjestelmän tietokannan tietojen siirtäminen uuden järjestelmään tietokantaan.

Nykyisestä järjestelmästä uuteen järjestelmään siirrettävät tiedot ovat seuraavat:

- Asiakkaat

- Lisenssit
- Lisenssien parametrit

Uusi järjestelmä toteutetaan toimimaan ilman Lotus Notes yhteyttä, toisin kuin nykyinen järjestelmä. Tämä tarkoittaa sitä, että uusia asiakkaita voidaan tehdä suoraan ilman asiakkaan perustamista johonkin toiseen järjestelmään. Nykyisen järjestelmän tietokannan tietorakenne on esiteltyä liittessä 1.

Nykyisessä tietokannassa on tallennettu asiakkaat tauluun Customers, lisenssit tauluun Licenses ja tietokantojen käyttötiedot tauluun Databases. Edelleen tietokannoilla voi olla omia parametreja, jotka on tallennettu tauluun Database_parameters.

Logiikka järjestelmässä toimii niin, että asiakastaululla ei ole mitään viitteitä lisensseihin. Lisenssien rakentaminen alkaa vasta Licenses-taulusta. Licenses-taulussa on lueteltuna kaikki olemassa olevat lisenssit. Mikäli lisenssiin on liitetty kanta, on se luotu Databases-tauluun ja siellä on linkki edelleen takaisin lisenssiin. Yhteen lisenssiin voi liittyä monta kanta. Mikäli kantaan on määritetty parametreja, on ne luotu Database_parameters-tauluun. Yhdellä kannalla voi olla vain yhden samannimiset parametrit.

Siirrettäessä nykyisen järjestelmän tietoja uuteen järjestelmään, ei tietojen kopiointi ollut suoraan mahdollista, johtuen uuden järjestelmän tietokannan rakenteen erilaisuudesta. Uudessa järjestelmässä tietorakenne rakennettiin uudelleen niin, että nykyisen kannan tietokannat kopioitiin uuteen kantaan suoraan lisensseiksi. Uudessa järjestelmässä yhteydet asiakkaan ja lisenssin välillä toteutetaan niin, että asiakkaalla on oma taulunsa Customer_licenses. Tämä taulu on mahdollista täydentää nykyisen järjestelmän avulla hyödyntäen sitä tietoa, että lisensseissä oli melkein aina kuvaus mihin asiakkaaseen lisenssi liittyi. Vertailussa käytettiin suoraan asiakkaan nimeä. Tämä vaati sen, että lisenssin kuvauksessa asiakkaan nimi oli kirjoitettu saman lailla kuin asiakkaan kannassa oli asiakkaan nimi kirjoitettuna. Eli siirrossa vertailtiin asiakkaan nimeä ja lisenssin nimeä, jolloin

tietty lisenssi oli mahdollista liittää asiakkaaseen. Nykyisestä tietokannasta tietojen siirtäminen uuteen tietokantaan tapahtui seuraavasti:

1. Kopioitiin nykyisestä kannasta kaikki asiakkaat uuteen kantaan.
2. Kopioitiin nykyisestä kannasta kaikki tietokannat uuteen kantaan lisensseiksi.
3. Luotiin uuteen kantaan parametrit parametri-tauluun nykyisestä kannasta.
4. Luotiin uuteen kantaan lisensseille lisenssiparametrit nykyisen kannan avulla.
5. Yhdistettiin uuteen kantaan asiakas tiettyyn lisenssiin nykyisen kannan avulla
6. Lisättiin käsin ne asiakaslisenssit, joille ei suoraan löytynyt nimiä vertailemalla lisenssiä. Tämä aiheutui siitä, että lisenssissä nimi saattoi olla lyhennettynä tai hieman eri lailla kirjoitettuna kuin asiakkaan kannassa asiakkaan nimi.

3.9 Asennusjärjestelmän tekeminen

Asennusohjelmia uudessa järjestelmässä tulee olemaan myös kaksi, aivan kuten nykyisessä järjestelmässä: yksi palvelimen asennukseen ja toinen työaseman asennukseen. Asennusohjelman lähdekoodi (NSIS-scripti) tullaan luomaan asiakkaittain OTO-ohjelman kautta asiakkaan version luonnin aikana. Version tekemisen aikana luotu asennuksen lähdekoodi käännetään asennusohjelmaksi NSIS-ohjelman mukana tulleella komentotyökalulla.

3.10 Toimitus- ja asennusprosessin vaiheet

Tässä kappaleessa käydään karkeasti läpi uuden toimitus ja asennusjärjestelmän toimitus- ja asennusprosessin vaiheet.

3.10.1 Version tekeminen uudella järjestelmällä

Uudella järjestelmällä version teko suoritetaan OTO-ohjelman kautta. Järjestelmän kautta pitää olla luotuna asiakas. Asiakas luodaan asiakkaan ylläpidon kautta. Asiakkaaseen voidaan liittää myös modifikaatioita eli asiakkaitaisia muutoksia, jotka toimitetaan asiakkaan version mukana. Asiakkaan ylläpidon kautta asiakkaaseen liitetään myös lisenssi ja jollei lisenssiä ole, se luodaan lisenssin ylläpidon kautta. Lisenssiin tulee oletuksena ne parametrit, jotka on määritetty oletusparametreiksi parametrien ylläpidon kautta. Lisenssille syötetään

vielä tarvittavat tiedot tietokannan yhteyden luomiseksi. Uudessa järjestelmässään lisenssit olivat itseasiassa tietokantojen kuvauksia.

Kun asiakas on olemassa ja siihen on liitetty lisenssi, voidaan edetä Version teko -vaiheeseen. Version teko -ruudulla on valittavissa ajoon liittyviä vaihtoehtoja, kuten suoritetaanko Powergen-ajo ja minne asiakasversio tallennetaan. Lopullinen rakenne tulee selville siinä vaiheessa, kun näyttö lopulta valmistuu.

Prosessi käynnistyy Powergen-ajolla. Powergen-ajon jälkeen asiakasversioon liitetään määritetyt lisenssitiedot ja asennusohjelmat ja siirretään versio hakemistoon, joka on asiakkaan tiedoissa määritetty valmiin version tallennuspaikaksi.

Version ajon jälkeen tuloksena on Oscar for Windowsin asennukseen tarvittava asennustiedosto.

3.10.2 Uuden asennusjärjestelmän palvelimen asennusohjelma

Uusi palvelimen asennusohjelma asentaa Oscar for Windowsin työaseman asennusohjelman yrityksen palvelimelta ajettavaksi sekä tekee tarvittavat kantapäivitykset ja erikoistoiminnot asiakkaan tietokantaan.

Palvelimen asennusohjelma tulee toimimaan seuraavia vaiheita noudattaen:

1. Kun asennusohjelma käynnistetään, kysytään kieli millä asennus halutaan suorittaa.
2. Toivotetaan käyttäjä tervetulleeksi Oscar for Windowsin palvelimen asennusohjelmaan
3. Näytetään lisenssiehdot.
4. Näytetään käyttäjälle lista mistä käyttäjä voi valita asennettavat osat.

Oletuksena kaikki on valittuna. Valittavina on seuraavia asioita:

- Laajennetaanko työasema-asennukseen tarvittava asennustiedosto
- Tehdäänkö kantapäivitys
- Näytetään listaus asiakaskohtaisista erikoistoiminnoista

5. Kysytään asennushakemiston sijainti, minne työaseman asennusta varten suoritettava asennustiedosto sijoitetaan.
6. Tehdään alustavia tarkistuksia järjestelmästä kuten :
 - Tarkistetaan onko Oraclen tietokantaan yhteyttä tietokantapäivityksiä varten, jotta ne voidaan ajaa kantaan. Asennusta ei kuitenkaan keskeytetä mikäli Oraclea ei löydetä, vaan annetaan käyttäjälle silti mahdollisuus edetä asennuksessa, mikäli käyttäjä niin valitsee. Poikkeavuudesta kuitenkin ilmoitetaan käyttäjälle. Mikäli kantapäivitys oli valittuna, tai valittuna oli erikoistoimintoja jotka vaativat kantaan yhteyttä, ne jätetään tekemättä.
7. Tehdään kantapäivitykset mikäli niiden ajamista ei poistettu
8. Ajetaan asiakkaan erikoistoiminnot mikäli niiden ajamista ei poistettu
9. Laajennetaan asennushakemistoon työaseman asennuksessa käytettävä asennustiedosto.
10. Näytetään loki tapahtuneista asioista ja tallennetaan loki asennushakemistoon

Onnistuneen asennuksen jälkeen Oscar for Windowsin käyttämä tietokanta palvelimella on päivitetty uutta versiota varten (mikäli kanta oli valittu päivitettäväksi), sekä Oscar for Windowsin työaseman asennuksessa käytettävä asennustiedosto on laajennettu asennushakemistoon.

Seuraavaksi on päivitettävä työasemille Oscar for Windowsin versiot vastaamaan päivitettyä kantaa. Aina ei kantapäivitys tietysti ole pakollinen, siksi sen tekeminen onkin käyttäjän valittavissa. Yleensä päivitys kuitenkin vaatii sekä kannan että Oscar for Windows -ohjelman version päivittämisen.

3.10.3 Uuden asennusjärjestelmän työaseman asennusohjelma

Uusi työaseman asennusohjelma asentaa Oscar for Windowsin työasemalle. Logiikka on samanlainen kuin nykyisessä järjestelmässä. Työaseman asennusohjelma tulee toimimaan seuraavia vaiheita noudattaen:

1. Asennusohjelma käynnistyessään kysyy kielen, millä asennus halutaan suorittaa. Tuetut kielet ovat suomi ja englanti.

2. Toivotetaan käyttäjä tervetulleeksi Oscar for Windowsin työaseman asennusohjelmaan.
3. Näytetään lisenssiehdot.
4. Kysytään sitä hakemistopolkua joka palvelimelta jaettuna, missä asennustiedostot sijaitsevat. Tämä vaihe määrittää valitun hakemiston rekisteriin talteen, jotta Oscar for Windows voi jatkossa käynnistyessään tarkistaa, onko palvelimelle tullut Oscar for Windowsista uudempaa versiota. Tarkistuksessa tullaan hyödyntämään build-numerotietoa, joka määritetään palvelimella sijaitsevaan pakettiin tiedoston ominaisuuksiin. Tätä build-numeroa vertaamalla Oscar for Windows -ohjelma osaa päivittää uudemman version tarvittaessa.
5. Näytetään listaus asennettavista tiedostoista ja muista optioista esimerkiksi tehdäanko pikakuvakkeet työpöydälle ja Windowsin Käynnistä / Ohjelmatvälikkoon.
6. Kysytään hakemisto, minne Oscar for Windows työasemalla asennetaan.
7. Näytetään yhteenveto valituista asioista ja suoritetaan asennus.
8. Kysytään haluaako käyttäjä nähdä lokitiedoston suoritetusta asennuksesta.

4 YHTEENVETO

Tutkintotyön tarkoituksena oli tuoda esille Oscar for Windows -toiminnanohjausjärjestelmän toimitus- ja asennusjärjestelmän asennuksessa tehtävien toimenpiteiden ongelmat ja määrittää uuden korvaavan toimitus- ja asennusjärjestelmän toimintalogiikka. Tilanne kuvattiin niin pitkälle kuin tällä hetkellä järjestelmän kehitys on edennyt. Samalla muodostettiin nykyisen järjestelmän toiminnasta kuvaus talteen, sillä aiemmasta järjestelmästä ei ollut dokumenttia olemassa. Uutta järjestelmää ei saatu vielä valmiiksi. Voidaan karkeasti arvioida ohjelmasta olevan valmiina n.50%.

Uutta järjestelmää kehitettäessä eteen tuli paljon uusia asioita ja opittavaa oli paljon. PowerBuilder-kehitysympäristöön tutustuin samalla kokoajan enemmän, kun työn tekemisessä etenin. Ohjelman käytössä nojauhin kuitenkin enemmän Oscar Softwaren työntekijöiden tietotaitoon, kuin siihen, että etsisin vastauksia Powerbuilderin manuaaleista tai Internetistä. Toki välillä näihinkin tietovarastoihin piti nojautua. NSIS-asennusjärjestelmän opiskelu oli aivan uutta, sen tutkimiseen ja käyttämiseen oli pakko käyttää lähinnä Internetiä ohjeiden etsimisessä mikäli ohjelman sisäisestä ohjeesta ei ollut riittävästi apua.

Samalla kun uutta järjestelmää kehitettiin pyrittiin ottamaan huomioon myös se, että uuden järjestelmän komponentteja olisi mahdollista käyttää myös muissa järjestelmissä mahdollisimman vähäisillä muutoksilla. Yksi tällainen ominaisuus on dynaaminen näyttöjen kokoaminen, joka mahdollistaa käyttäjäkohtaisten näyttöjen rakentamisen.

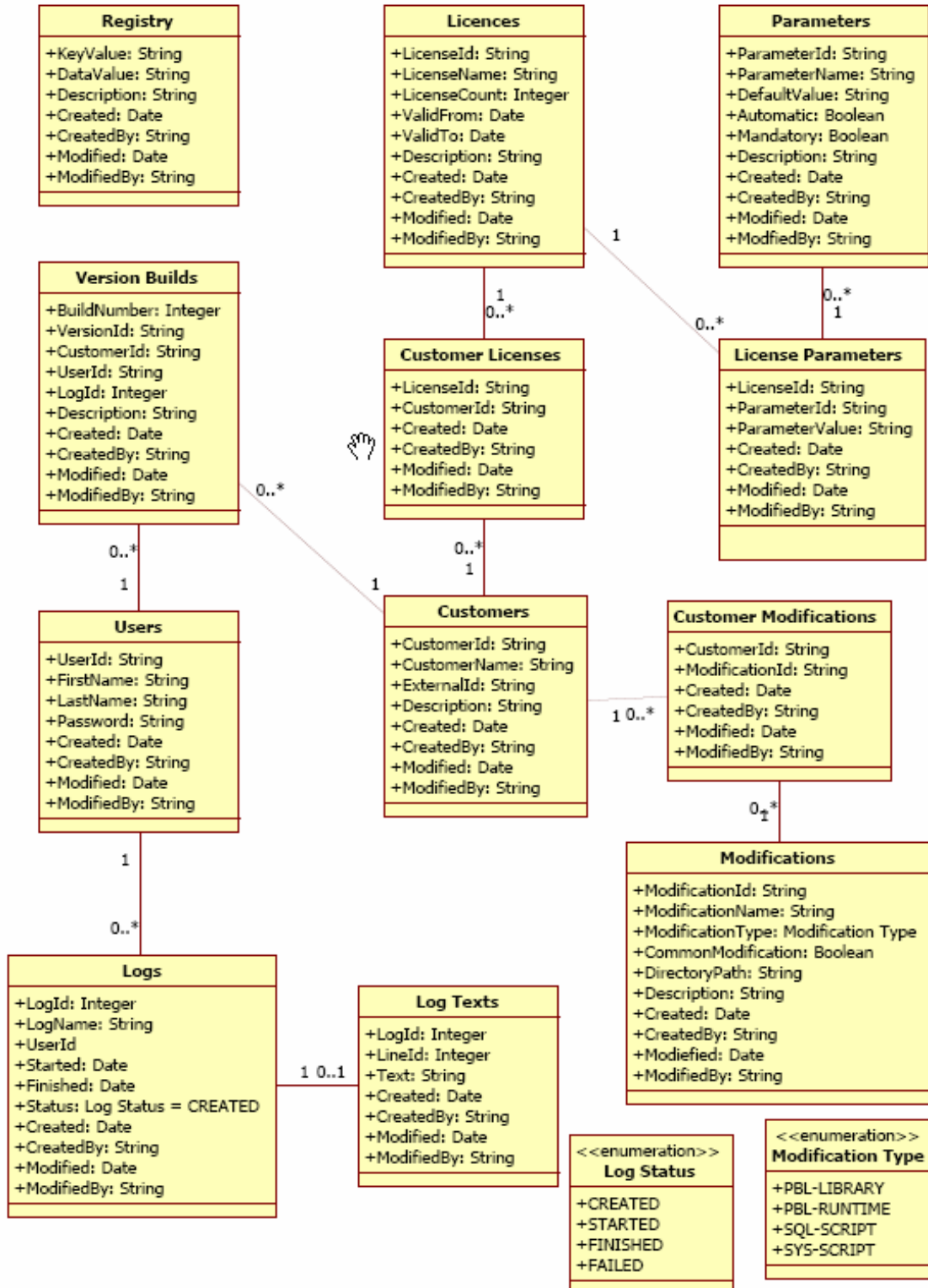
Tutkintotyötä tehdessä sain paljon arvokasta tietoa sen tekemiseen käyttämästäni työkaluista, jota voin hyödyntää myös Oscar for Windows -ohjelman kehityksessä sekä muissa projekteissa. Oscar Software sai puolestaan asennus- ja toimitusjärjestelmän, jonka tekemisessä on otettu huomioon helpompi muokattavuus ja ylläpito kuin nykyisessä järjestelmässä on mahdollista.

LÄHTEET

Sähköiset lähteet

- 1 Oracle Community. [WWW-sivu]. [Viitattu 14.4.2007]
http://www.oracle.com/technology/obe/2day_dba/schema/schema.htm
- 2 Oscar for Windows.[WWW-sivu]. [Viitattu 14.4.2007]. <http://www.oscar.fi>
- 3 Sybase DataWindow® Programmer's Guide. [WWW-sivu]. [Viitattu 14.4.2007].
<http://manuals.sybase.com/onlinebooks/group-pb/pbg0900e/dwprgug/>
- 4 Oracle Database 10g. [WWW-sivu]. [Viitattu 14.4.2007].
<http://www.oracle.com/database/index.html>
- 5 ToadSoft. [WWW-sivu]. [Viitattu 14.4.2007].
http://www.toadsoft.com/toad_oracle.htm
- 6 NullSoft Scriptable Install System. [WWW-sivu]. [Viitattu 14.4.2007].
http://nsis.sourceforge.net/Main_Page

Liite 1 Uuden toimitus- ja asennusjärjestelmän tietokannan UML-kuvaus



Liite 2 Opak-ohjelman kuvaus

Oscar for Windowsin kääntäminen ajettavaan muotoon Opak-ohjelman kautta vaatii seuraavien tiedostojen olemassaoloa:

- Opak.exe
- Opak.ini
- Customers.ini
- Libs.ini
- Releases.ini

Opak.exe

Opak.exe on ohjelma, jolla koko käännösprosessin hallinnointi tapahtuu. Ohjelma hyödyntää edellä mainittuja INI-tiedostoja toiminnoissaan. Ohjelma kerää kaikki tarvittavat tiedostot OfW:n perusversion rakentamiseksi, ja mikäli asiakkaalla on olemassa erikoiskirjastoja jotka sisältävät asiakasta varten kehitettyjä ominaisuuksia, ne otetaan ohjelman kääntämiseen mukaan. Kun OfW:n versio on käännetty, paketoidaan se yhdeksi tiedostoksi johon vielä liitetään ohjelman ajoa varten tarvittavia DLL-tiedostoja sekä myös OfW:n ohjetiedostot jotka ovat PDF-muodossa. Kaikki tiedostot sijaitsevat lähiverkossa ja näiden hakemistojen sijainti on määritetty ohjelman luettavaksi erillisestä Opak.ini-tiedostosta. Ohjelma päättelee asiakaskohtaisten tiedostojen sijainnin asiakasnumeron perusteella. Versioita voi laittaa Opakilla tehtäväksi maksimissaan neljä yhtäaikaaisesti. Kaikki ajot suorittavat tiedostojen kopioinnin ajoa varten väliaikaishakemistoihin, mutta vain ensimmäisenä käynnistetty alkaa suorittamaan ohjelman kääntämistä ajomuotoon. Muut odottavat jonossa vuoroaan.

Opak ohjelmaa käyttäessäni voin todeta ohjelma toimineen hyvin ja sen olleen helppokäyttöinen, kunhan INI-tiedostot on määritelty oikein. Ohjelma on siis suorittanut hyvin sille määritetyt tehtävät.

Opak.ini

Opak.ini:stä Opak.exe lukee asetuksia, joita se käyttää suorittaessaan OfW:n version tekemistä. Tiedostoon on määritelty seuraavat tiedot:

- version rakentamiseen tarvittavien tiedostojen juurihakemiston sijainti
- luettelo sähköpostiosoitteista, joihin ohjelma voidaan määrittää lähettämään tieto version valmistumisesta
- sähköpostipalvelimen osoite, jonka kautta Opak lähettää tiedon valmistuneesta OfW:n versiosta
- käännöksessä käytettävän Powergen-ohjelman sijaintihakemisto
- Powergen-ohjelmaa varten määritellyt lisäparametrit

Customers.ini

Customers.ini:ssä on lueteltuna kaikki asiakkaat asiakasnumeroineen. Tiedoston avulla Opak.exe osaa hakea oikeasta hakemistosta asiakaskohtaiset kirjastot ja tehdä siirtää valmistuneen version oikeanimiseen hakemistoon asiakasnumeron perusteella. Mikäli järjestelmään on tarkoitus liittää uusi asiakas, tapahtuu se lisäämällä asiakkaan asiakasnumero ja nimi tähän tiedostoon sen jälkeen, kun Lotus Notes -ohjelmalla on perustettu asiakkaan tiedot asiakastietoihin.

Libs.ini

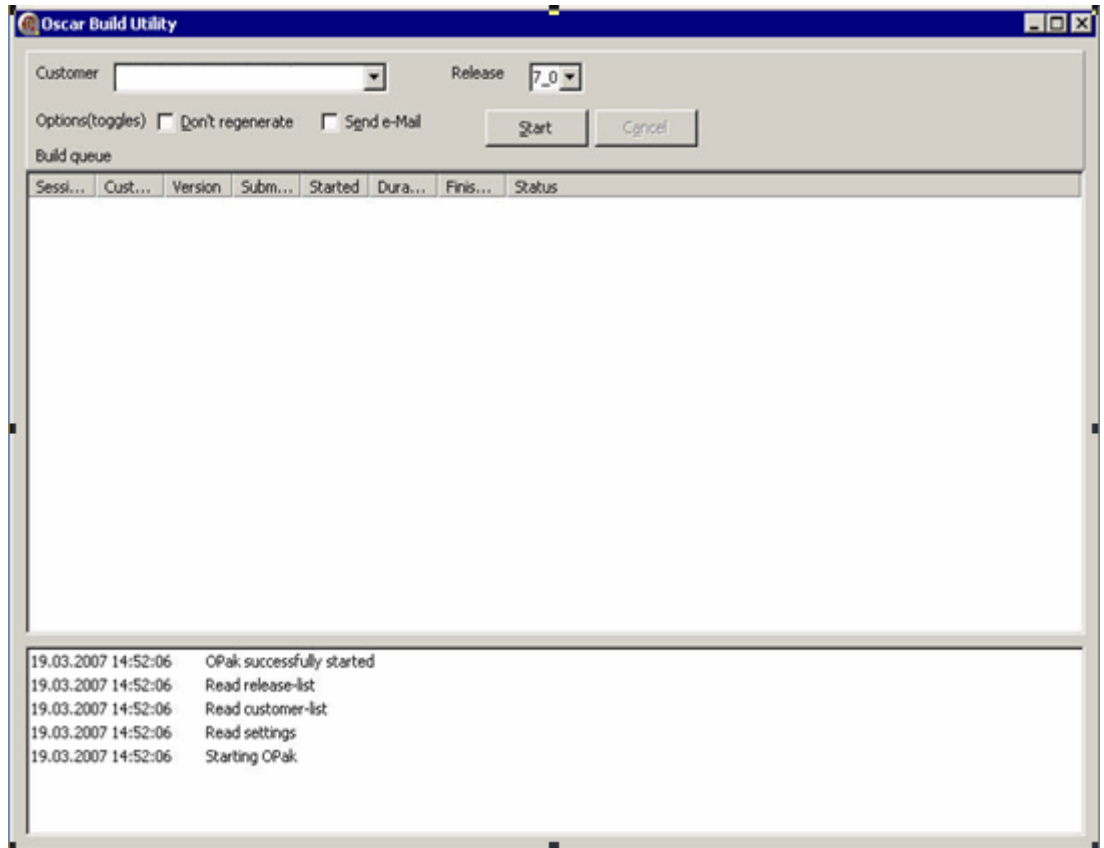
Libs.ini kertoo Opak.exe:lle mitä kirjastoja version tekemisessä käytetään Powergen ajon käynnistyessä.

Releases.ini

Releases.ini-tiedostossa on määriteltynä Opak.exe:n kautta käännettäväksi valittavat versiot Oscar for Windows -ohjelmasta.

Liite 3 Oscar for Windowsin version kääntäminen Opak-ohjelmalla

Tässä kappaleessa käydään läpi version tekeminen Opak-ohjelmalla. Ohjelman käynnistyttyä käyttäjälle avautuu kuvan 1 mukainen näkymä:



Kuva 1 Opak-ohjelman päänäkymä

Oscar for Windowsin versio tehdään Opak-ohjelmalla seuraavia toimenpiteitä seuraten:

1. Valitaan asiakas alasvetovalikosta kohdasta Customer.
2. Valitaan ajettava versio alasvetovalikosta Release, uusin versio 7_0.
3. Mikäli halutaan lähettää sähköpostia version valmistumisesta, valitaan kohta Send e-Mail.
4. Painetaan Start-nappia. Mikäli kohta ” Send e-Mail” oli valittuna, ohjelma näyttää ennen ajon alkamista käyttäjälle listan, josta valitaan kenelle sähköpostia version valmistumisesta lähetetään.

Ohjelma suorittaa version tekemisen väliaikaishakemistossa. Version tekemisessä ohjelma etenee automaattisesti seuraavissa vaiheissa (tässä esimerkissä ohjelma kääntää version 7.0)

1. Kopioidaan Opak.ini:ssä määritetyn juuripolun alta alihakemistosta `\Client\Libs.7_0\` kaikki tiedostot väliaikaishakemistoon. Nämä tiedostot sisältävät Oscar for Windowsin perusversion kääntämättömänä ja tulevat aina mukaan kaikkiin Opak.exe:llä ajettuihin versioihin.
2. Kopioidaan asiakkaan erikoiskirjastot asiakasnumeron perusteella hakemistosta `\Client\Special\asiakasnumero.7_0\`.
3. Kopioidaan version kääntämisessä tarvittavat grafiikkatiedostot hakemistosta `\Build\Images.7_0\`.
4. Kopioidaan väliaikaiseen hakemistoon versioon liitettävät muut tiedostot kuten Oscar for Windowsin manuaalit, ohjelman ajon aikana tarvittavat DLL-tiedostot sekä muut lisätiedostot hakemistosta `Client\Common.7_0\`.
5. Kopioidaan väliaikaiseen hakemiston alle Client-hakemistoon asennusohjelma Client-asennusta varten hakemistosta `\Client\Install.7_0\`.
6. Käännetään versio kerätyistä tiedostoista.
7. Käännetystä versiosta paketoidaan yksi tiedosto nimeltään Client.zop, joka sijoitetaan väliaikaishakemiston alle hakemistoon Client. Tähän tiedostoon sisällytetään käännetyt kirjastot ja muut oleelliset lisätiedostot kuten DLL- ja manuaali-tiedostot.
8. Pakataan väliaikaisen hakemiston alla olevassa hakemistossa Client kaikki tiedostot väliaikaishakemiston juureen tiedostoon Client.dat.
9. Hakemistosta `Server\Common.7_0\` kopioidaan Server-asennustiedosto sekä edellisessä vaiheessa luotu Client.dat-tiedosto hakemistoon `\Distribution\asiakasnumero.7_0\`.
10. Ohjelma poistaa väliaikaiset tiedostot.
11. Version tekeminen on Opak-ohjelman osalta valmis. Ohjelmaan liitetään seuraavaksi lisenssi-tiedosto ohjelmalla LiMe.

Uusi järjestelmä tulee seuraamaan Opakin jalanjälkiä version tekemisessä. Edellä mainituista vaiheista turhiksi tulee muodostumaan asennusohjelmien liittäminen versioon. Uudessa järjestelmässä asennusohjelmat tehdään uusiksi ja pohdinnan alla on myös luovutaanko nykyisestä tiedostojen paketoinnista. Osittain luopumista vastaan puhuu se seikka, että silloin kun Oscar for Windowsin nykyinen version tarkistus suoritetaan työasemalla Oscaria käynnistettäessä, niin tämä

tarkistusohjelma perustuu siihen, että serveriltä löytyy Client.zop niminen paketti, joka tarpeen vaatiessa puretaan työasemalle ja näin siis myös päivitetään Oscarin versio. Mikäli paketointi halutaan suorittaa uudella logiikalla, pitää myös Oscarin käynnistyksessä ajettavaa tarkistusohjelmaa muuttaa käsittelemään uuden asennusohjelman tekemä paketointi.

Liite 4 LiMe -ohjelman kuvaus ja käyttö

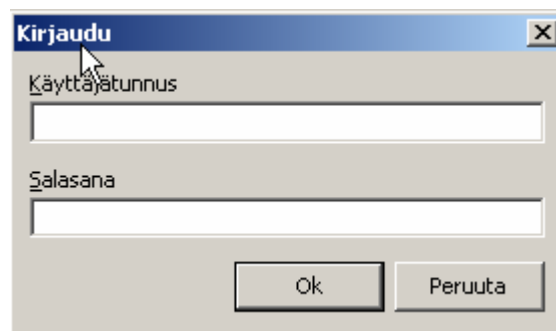
Lime voidaan asentaa mille koneelle tahansa, missä on asennettuna myös Lotus Notes. Lotus Notesia Lime käyttää asiakkaan tietojen hakemista varten. LiMe-ohjelman ajamiseksi pitää olla myös yhteydessä Oscar Softwaren sisäverkkoon. LiMe lisensoi Oscar for Windowsin version minkä Opak on tehnyt liittämällä siihen lisenssitiedoston ja paketoimalla kaiken tiedostoon setup.pak, jonka taas palvelimen asennusohjelma osaa purkaa. Lisäksi mukaan liitetään avaintiedosto, joka sisältää avaimen millä asennus voidaan käynnistää. LiMellä tehty lisenssitiedosto sisältää luettelon niistä tietokannoista ja kantoihin liittyvistä parametreista, joihin kyseiselle asiakkaalle toimitetulla Oscar for Windows -ohjelmalla voidaan ottaa yhteys. Tietokantoja voi olla yksi tai useampia.

LiMe lukee käynnistysasetuksensa tiedostosta lime.ini, joka sijaitsee samassa hakemistossa kuin ohjelma itse. Tiedostossa määritetään seuraavat tiedot LiMen ajamista varten

1. LiMen käyttämän tietokannan sijainti
2. Opakin tekemien versioiden juurihakemiston sijainti
3. Asiakastietojen hakemiseksi käytettävän Lotus Notesin tietokannan sijainti

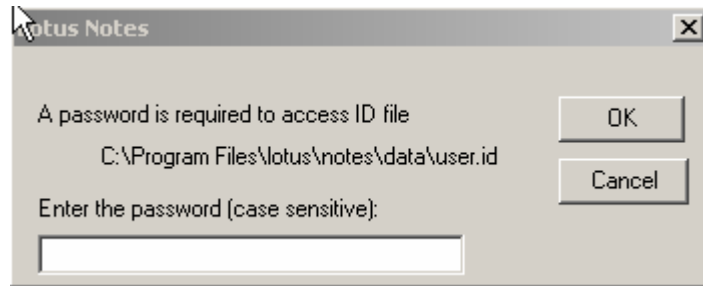
Seuraavaksi käydään läpi ohjelman lisensoiminen tekemällä uusi lisenssi olemassa olevalle asiakkaalle LiMen avulla.

1. Käynnistetään ohjelma ja syötetään käyttäjätunnus ja salasana (kuva 2)



Kuva 2 LiMe-ohjelma kirjautumisruutu

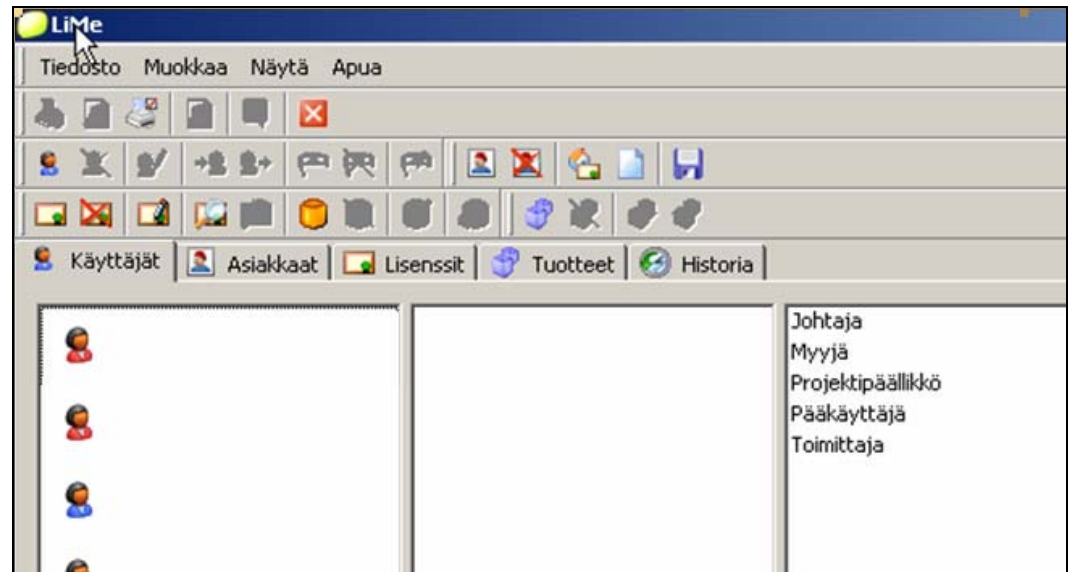
2. Syötetään vielä tunnus liittyen Lotus Notes -ohjelmaan asiakastietojen hakemista varten (kuva 3)



Kuva 3 Lotus Notes -kirjautumisruutu

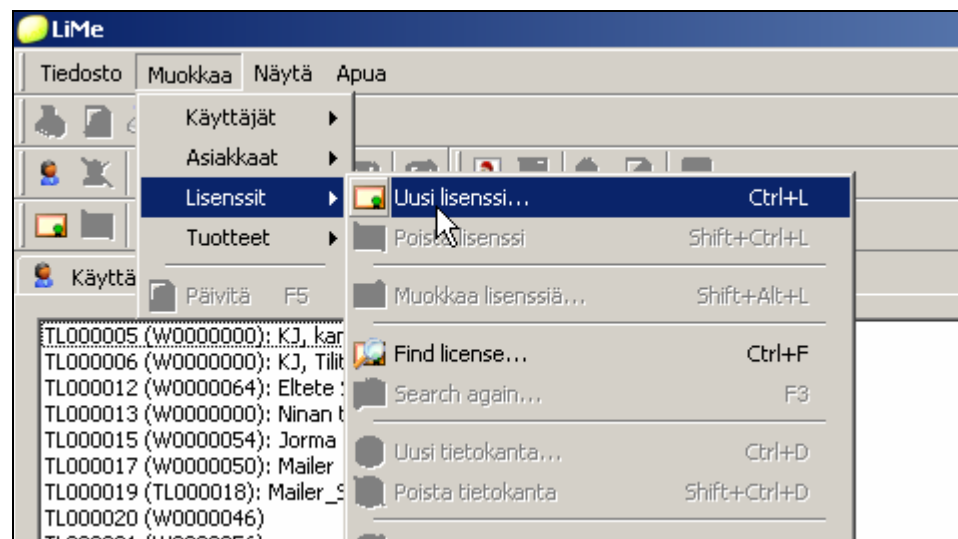
3. Ohjelmassa on seuraavat välilehdet (kuva 4)

- Käyttäjät
Käyttäjät-välilehden kautta määritetään ohjelman käyttäjät ja heidän roolinsa. Käytössä näistä on tällä hetkellä Pääkäyttäjä- sekä Projektipäällikkö-roolit, sillä muille ei ole juurikaan käyttöä. Pääkäyttäjä roolikin sinänsä riittää, koska sillä on kaikki oikeudet ohjelmassa.
- Asiakkaat
Asiakkaat-välilehdellä näkyy asiakkaat jotka on lisätty Lotus Notes -ohjelmaan josta asiakkaiden tilanne päivittyy myös LiMe-ohjelmaan.
- Lisenssit
Lisenssit välilehdellä näkyy lisenssit
- Tuotteet
Tämä välilehti ei ole juurikaan käytössä enää. Sieltä on voitu aikanaan vaihtaa ohjelma mitä lisensoidaan, mutta nykyään on enää käytössä ohjelman Oscar for Windows lisensointi.
- Historia
Historia-välilehdellä näkyy historiatiedot ohjelmalla tehdyistä toiminnoista.



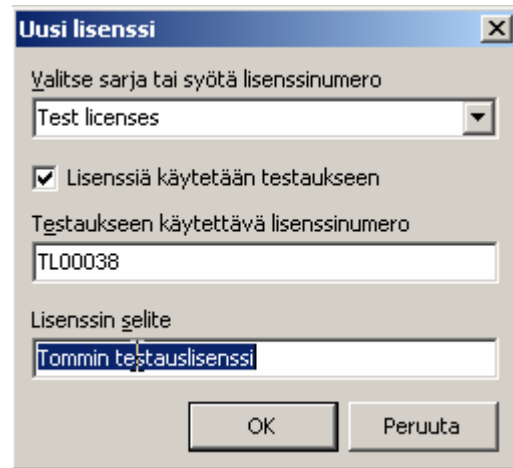
Kuva 4 LiMe-ohjelman päänäyttö

4. Valitaan valikosta Muokkaa / Lisenssit / Uusi lisenssi (kuva 5):



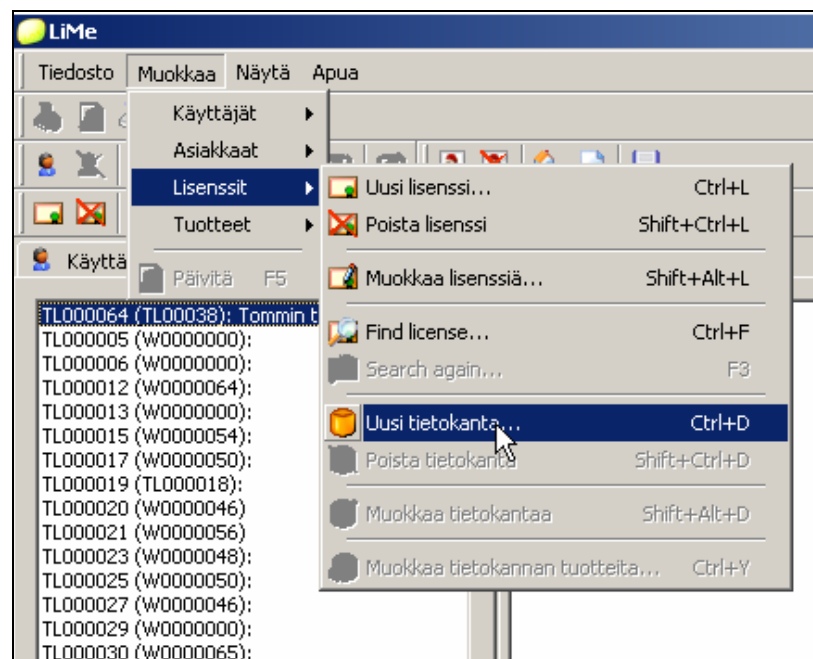
Kuva 5 Uuden lisenssin tekeminen

5. Syötetään tiedot avautuneelle ruudulle (kuva 6). Ohjelmaan on mahdollista tehdä testauslisenssejä, sekä virallisia lisenssejä. Testauslisenssejä voidaan käyttää, kuten nimikin kertoo, testaukseen. Virallinen versio Oscar for Windowsista toimitetaan virallisella lisenssillä. Kuvan 6 mukaisesti tehdään testilisenssi. Kirjoitetaan ”Valitse sarja tai syötä lisenssinumero”-kohtaan ”Test licenses”, valitaan kohta ”Lisenssiä käytetään testaukseen”, kirjoitetaan kohtaan ”Testaukseen käytettävä lisenssinumero” kirjaimet ”TL”, jolloin ohjelma itse hakee seuraavan vapaan numeron kirjainten TL perään ja lopuksi annetaan ”Lisenssin selite”-kohtaan lyhyt selite lisenssistä.



Kuva 6 Lisenssin tietojen syöttäminen

6. Lisenssin alustavan luonnin jälkeen siihen liitetään yksi tai useampi tietokanta. Liittäminen tapahtuu menemällä ohjelmassa Lisenssit-välilehdelle ja valitsemalla juuri lisätty lisenssi. Tämän jälkeen valitaan valikosta Muokkaa / Lisenssit / Uusi tietokanta (kuva 7)



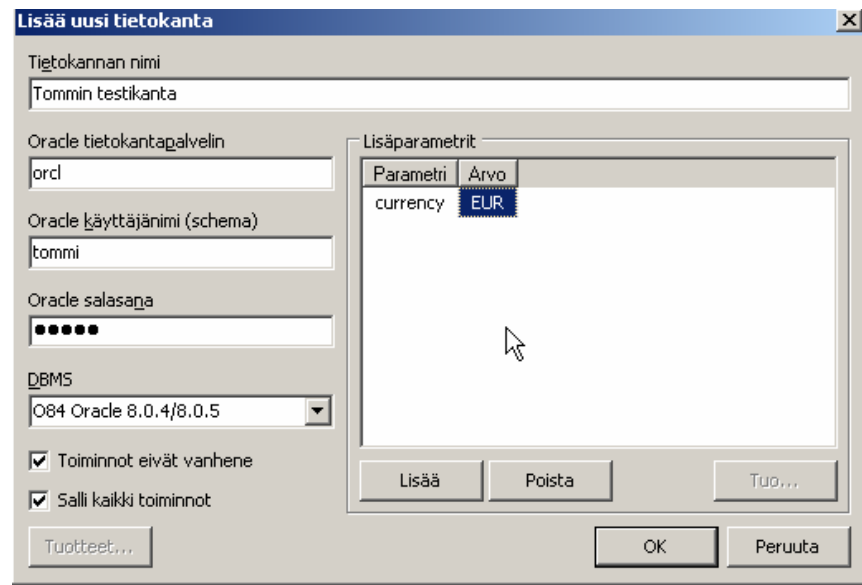
Kuva 7 Tietokannan liittäminen lisenssiin

7. Avautuu kuvan 8 mukainen ikkuna. Kirjoitetaan seuraavat tiedot:
- Tietokannan nimi -kohdassa on jo valmiina lisenssin selite.
 - Oracle tietokantapalvelin -kohtaan kirjoitetaan palvelimen nimi
 - Oracle käyttäjänimi -kohtaan kirjoitetaan Oraclen tietokantascheman nimi

- Oracle salasana -kohtaan kirjoitetaan salasana schemaan
- DBMS-kohtaan valitaan alavetovalikosta tai kirjoitetaan itse Oraclen versio, mikä on käytössä Oscar for Windows-ohjelman kanssa
- Lisätään tarvittavia lisäparametreja Lisäparametrit-kohtaan.

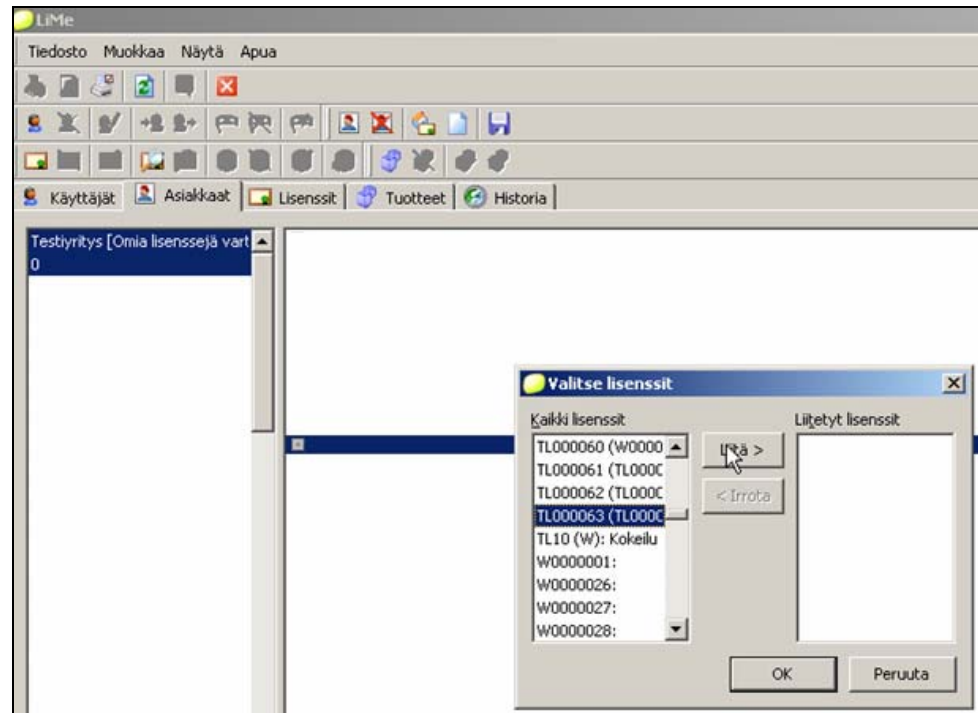
Mahdollisia parametreja ovat :

- Currency, kertoo kannassa käytettävän valuutan tyyppin
- NumberUsers, kertoo kuinka monta käyttäjää voi olla yhtä aikaa kannassa kiinni
- Special1; Special2; Special3, erikoistoimintoja jotka näkyvät OfW-ohjelmassa mikäli ne on määritetty



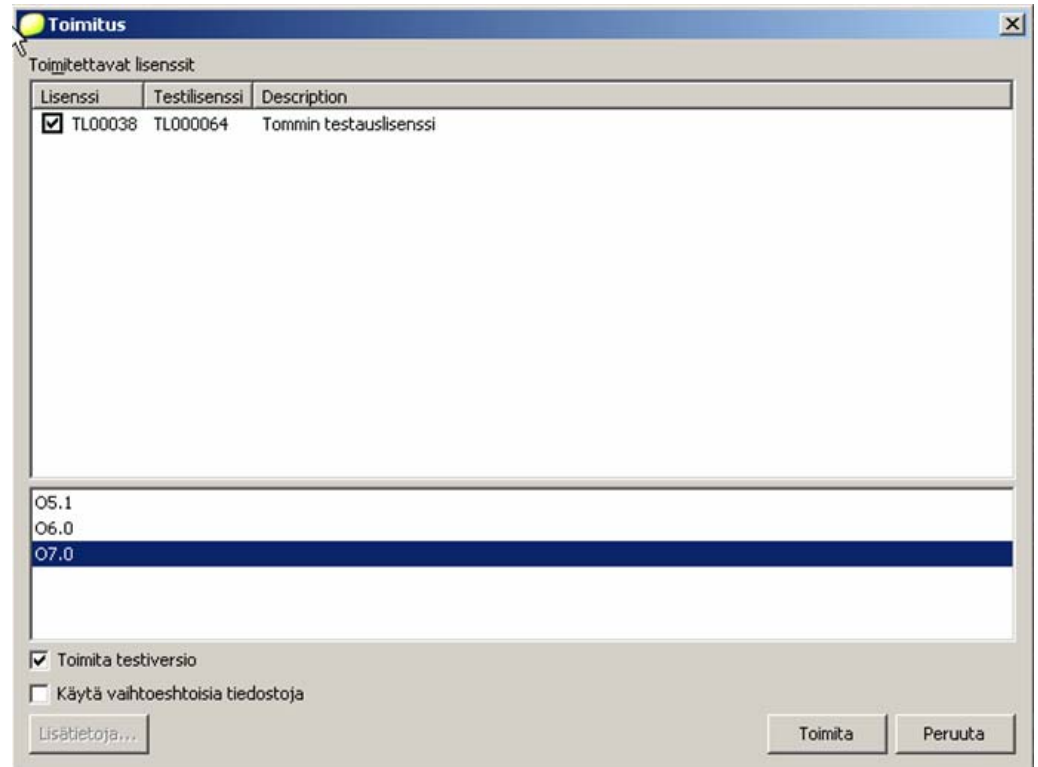
Kuva 8 Tietokannan parametrien määrittäminen

8. Mikäli halutaan liittää toinen tietokanta lisenssitiedostoon, toistetaan edellinen vaihe.
9. Kun haluttu määrä tietokantoja on liitetty, määritetään seuraavaksi lisenssi jonkin henkilön alle. Mennään Asiakas-välilehdelle ja valitaan haluttu asiakas. Kaksoisnäpäytetään oikealla puolella näyttöä olevaa henkilöä (tässä kuvaa on muokattu niin, että henkilön nimi on poistettu ja näkyy vain tumma rivi), jolloin aukeaa lista lisensseistä, jotka voidaan liittää henkilöön (kuva 9).



Kuva 9 Lisenssin liittäminen

10. Kun haluttu lisenssi on liitetty, toimitetaan versio painamalla ruudulla näkyvää levykkeen kuvaa, jolloin avautuu kuvan 10 mukainen ikkuna. Ruudulla voidaan näyttää joko testilisenssit tai viralliset lisenssit muuttamalla valintaa kohdasta ”Toimita testiversio”. Ruudulta voidaan valita toimitettavaksi tietty versio Oscar for Windowsista, mutta yleensä toimitetaan aina viimeisin versio, eli tällä hetkellä versio 7.0.



Kuva 10 OfW:n version toimittaminen

11. Kun lisenssi on valittu, painetaan ”Toimita”-nappia, jolloin ohjelma vielä kysyy käyttäjältä minne versio tallennetaan (kuva 11). Käyttäjän painettua OK-nappia, ohjelma kopioi verkosta asiakkaan tiedostot, jotka Opak on muodostanut ja liittää pakettiin vielä lisenssin. Lopuksi tiedostot pakataan vielä yhdeksi tiedostoksi nimeltään setup.pak, jonka palvelimen asennusohjelma osaa purkaa.



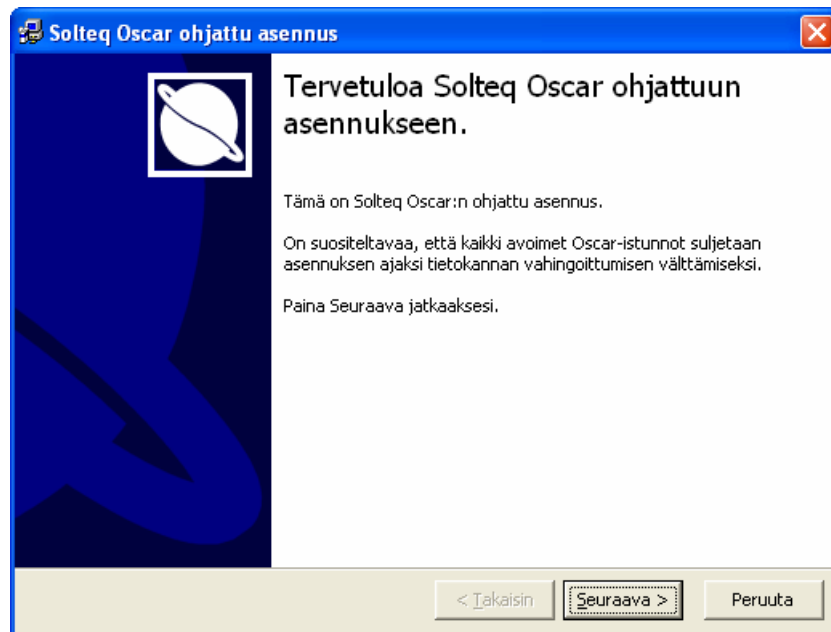
Kuva 11 OfW:n version tallennushakemiston määrittäminen

12. Oscar for Windowsin versio on nyt valmis asennettavaksi.

Liite 5 Palvelimen asennusohjelma

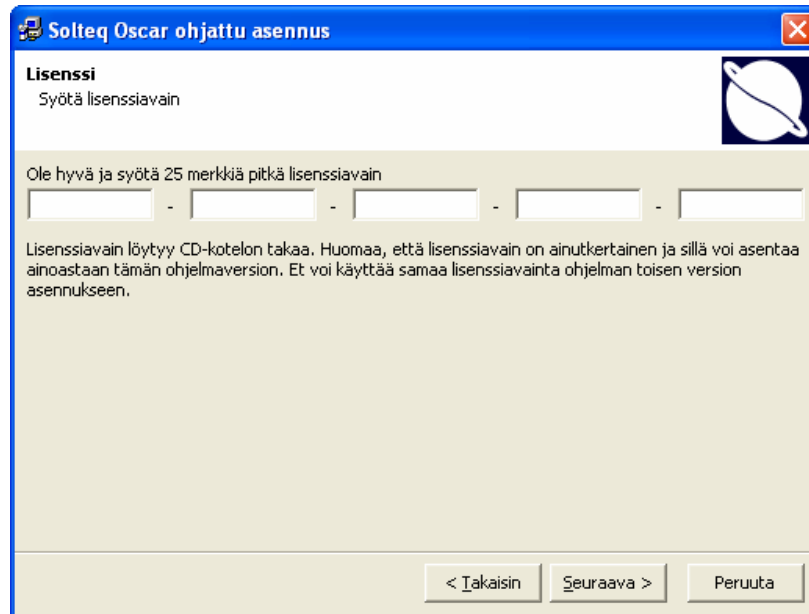
Palvelimen asennusohjelma ajetaan siellä, missä Oscar for Windowsia varten asennetut tietokannat sijaitsevat. Ohjelma kopioidaan johonkin hakemistoon, esimerkiksi koneen C:-asemalle hakemistoon ofw_asennus. Hakemistosta löytyy tiedosto keyfile.txt, missä on asennuksen alussa tarvittava avain, jotta ohjelman asennusta voidaan jatkaa. Asennus etenee seuraavasti

1. Palvelimen asennusohjelma käynnistetään tiedostosta Setup.exe. Ohjelma toivottaa käyttäjän tervetulleeksi kuvan 1 mukaisesti. Painetaan Seuraava-nappia.



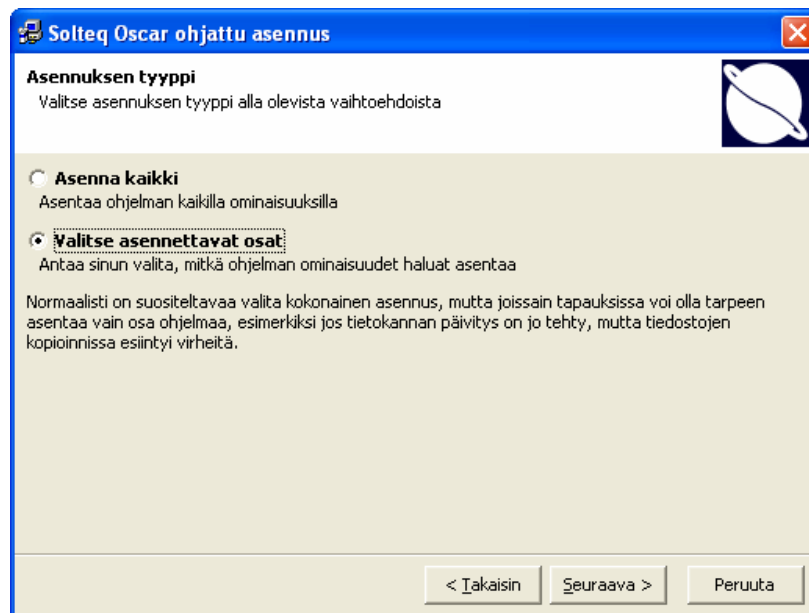
Kuva 1 OfW:n palvelimen asennusohjelman käynnistysruutu

2. Ohjelma kysyy kuvan 2 mukaisesti lisenssiavainta. Syötetään se ja painetaan Seuraava-nappia.



Kuva 2 OfW:n palvelimen asennusohjelman lisenssiavaimen syöttöruutu

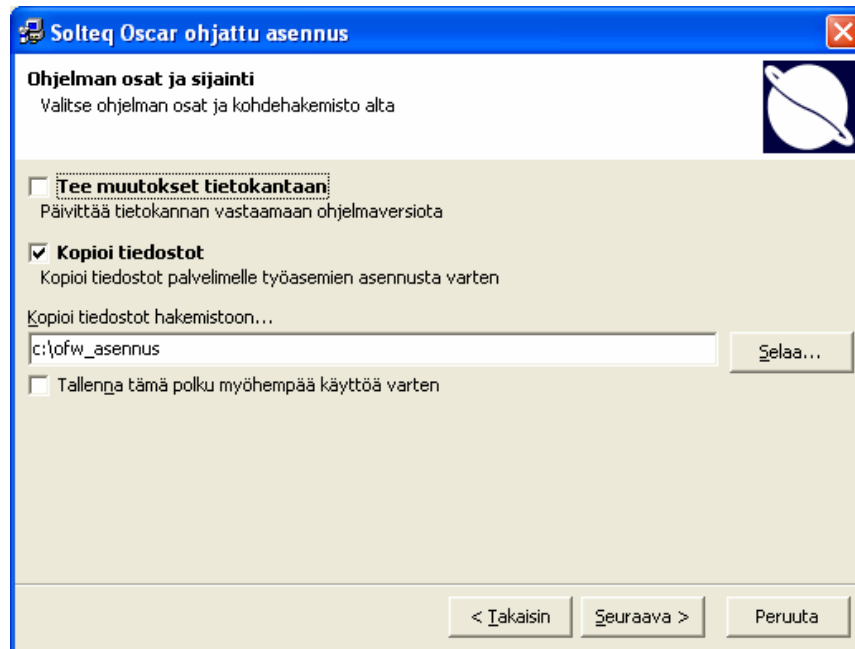
3. Ohjelma kysyy kuvan 3 mukaisesti asennetaanko kaikki osat vai valitaanko asennettavat osat. Tästä on jouduttu tällä hetkellä aina valitsemaan vaihtoehto ”Valitse asennettavat osat”, johtuen asennusohjelman yhteensopimattomuudesta nykyisen Oscar for Windowsin tietokantajärjestelmän kanssa. Valitaan siis alempi vaihtoehto ja painetaan Seuraava-nappia.



Kuva 3 OfW: palvelimen asennusohjelman asennustavan valitsemisruutu

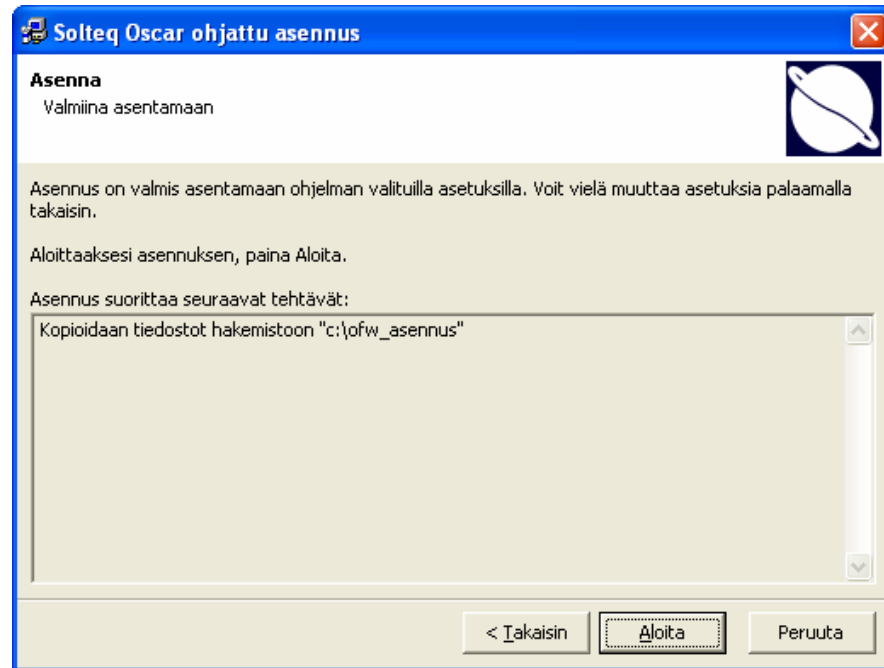
4. Mikäli aiemmassa kohdassa olisi valittu ”Asenna kaikki”-vaihtoehto, niin myös tietokantapäivitykset olisi ajettu. Tämä on yksi ominaisuus mikä ei tällä

hetkellä asennuksessa toimi, koska kuten aiemmin todettiin, ohjelma ei ole yhteensopiva nykyisen kantajärjestelmän kanssa. Siksi poistetaan kuvan 4 mukaisesti valinta kohdasta ”Tee muutokset tietokantaan” ja valitaan vain kohta ”Kopioi tiedostot”. Vielä valitaan hakemisto minne tiedosto asennetaan. Tähän määritetään yleensä sama hakemisto mistä palvelimen asennusohjelmakin käynnistettiin, toki voidaan määrittää muukin hakemisto. Voidaan myös merkitä kohta ”Tallenna tämä polku myöhempää käyttöä varten” valituksi, jolloin tehtäessä uutta palvelimen asennusta ohjelma muistaa edellisestä asennuskerrasta saman asennuspaikan uutta asennusta varten. Seuraavaksi painetaan jälleen Seuraava-nappia.



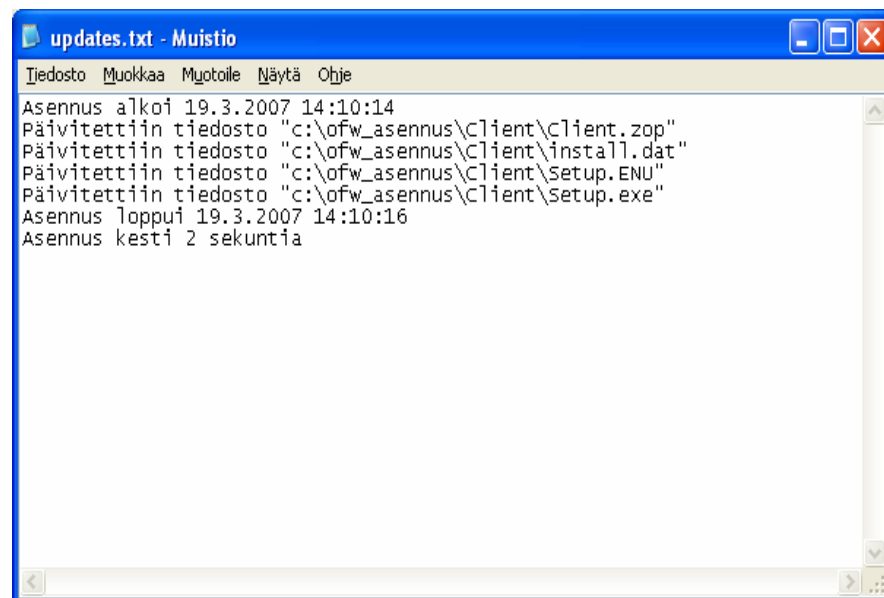
Kuva 4 OfW:n palvelimen asennusohjelman asetusten määrittämisruutu

5. Ohjelma näyttää vielä yhteenvedon ajettavasta prosessista kuvan 5 mukaisesti. Asentaminen aloitetaan painamalla Aloita-nappia.



Kuva 5 OfW:n palvelimen asennusohjelman valintojen yhteenvetoruutu

- Asennuksen jälkeen ohjelma kysyy haluaako käyttäjä nähdä lokitiedoston tapahtumista ja mikäli käyttäjä vastaa myöntävästi ohjelma näyttää lokitiedoston (Kuva 6)



Kuva 6 OfW:n palvelimen asennusohjelman lokitiedoston tiedot

- Lokitiedostosta nähdään, että palvelimen asennusohjelma teki määritettyyn asennushakemistoon työaseman asennusta varten tiedostot "Client"-hakemiston alle. Lokitiedostosta poiketen asennusohjelma tekee "Client"-nimisen hakemiston alle myös "User"-nimisen hakemiston, jonne se sijoittaa Oscar for

Windowsin käyttämän lisenssitiedoston License.clf. Tämän hakemiston kautta on jatkossa mahdollista levittää kaikille työasemille päivityksen aikana myös muita tiedostoja, sillä Oscar for Windows kopioi käynnistyessään täältä hakemistosta tiedostot omaan hakemistoonsa.

Liite 6 Työaseman asennusohjelma

Työaseman asennuksen tekemiseksi jaetaan verkkoon palvelimen asennusohjelman tekemä hakemisto "Client" niille käyttäjille, joiden on tarkoitus asentaa Oscar for Windows -ohjelma koneellensa. Kun jako on tehty ja oikeudet annettu kyseiseen hakemistoon, voidaan työaseman asennusohjelma käynnistää.

Ohjelman suoritus etenee seuraavasti

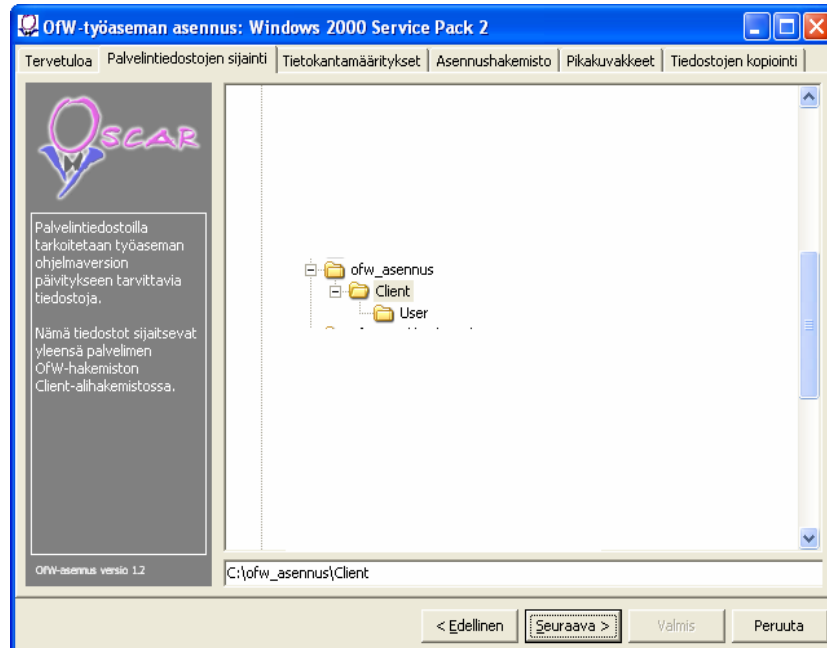
1. Käynnistetään verkkojaon kautta Client-hakemistossa sijaitseva Setup.exe-tiedosto jolloin ruudulle ilmestyy kuvan 1 mukainen ikkuna. Ohjelma toivottaa käyttäjän tervetulleeksi Oscar for Windowsin asennukseen ja vaatii käyttäjää tutustumaan ruudulla näkyviin käyttöehtoihin ennen asennuksen jatkamista. Ohjelmassa voi vaihtaa asennusvaihetta klikkaamalla suoraan eri tab-sivuja ylhäältä tai painamalla ohjelman alaosassa sijaitsevia nappeja. Tässä vaiheessa merkataan kohta "Olen tutustunut ohjelman käyttöehtoihin ja hyväksyn ne" valituksi ja painetaan Seuraava-nappia.



Kuva 1 OfW:n työaseman asennusohjelman käynnistysruutu

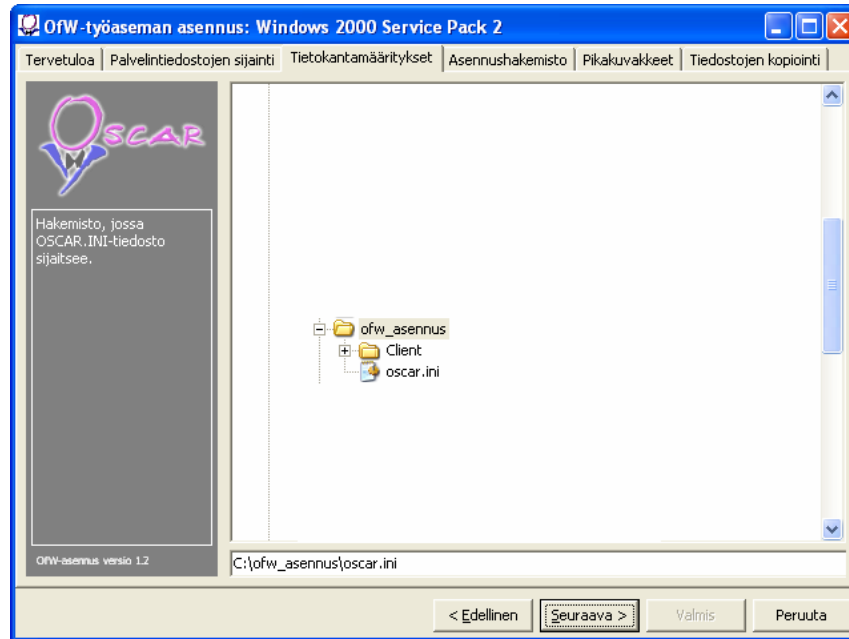
2. Ohjelma pyytää käyttäjää kuvan 2 mukaisesti määrittämään palvelintiedostojen sijainnin. Palvelintiedostojen sijainti on sama kuin mistä työaseman

asennusohjelman käynnistävä Setup.exe-tiedosto käynnistettiin. Hakemiston määrittämisen jälkeen painetaan Seuraava-nappia.



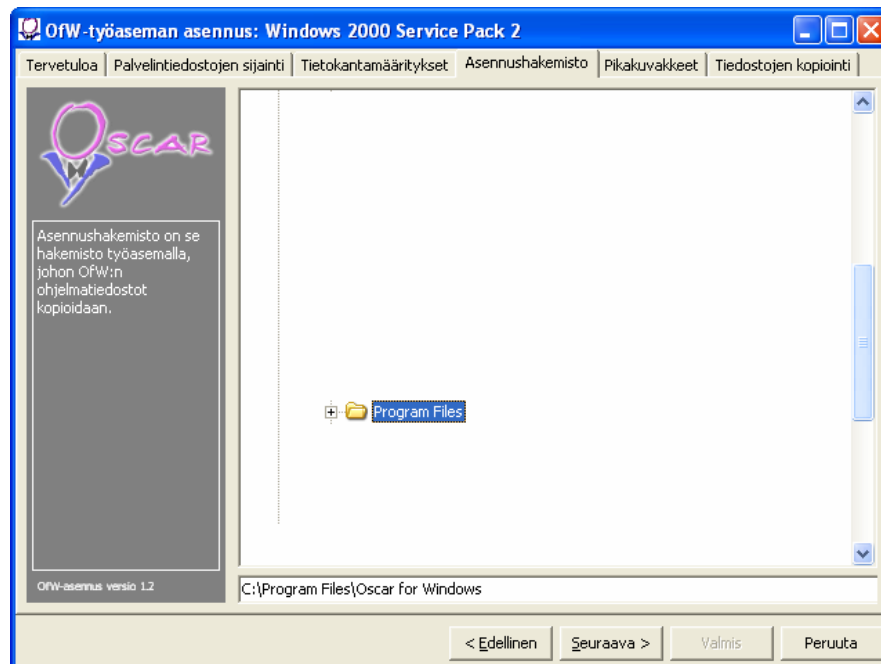
Kuva 2 OfW:n työaseman asennusohjelman palvelintiedostojen sijainnin määrittämisruutu

- Ohjelma kysyy Oscar.ini-tiedoston sijaintia kuvan 3 mukaisesti. Tämä on tällä hetkellä turha vaihe, mutta johtuen ohjelman muokkaamattomuudesta sitä ei voida ohittaa, joten ohjelmaa varten pitää tehdä tyhjä Oscar.ini-tiedosto ja määrittää hakemisto missä se sijaitsee. Tiedoston tekemisen ja sijainnin määrittämisen jälkeen painetaan Seuraava-nappia.



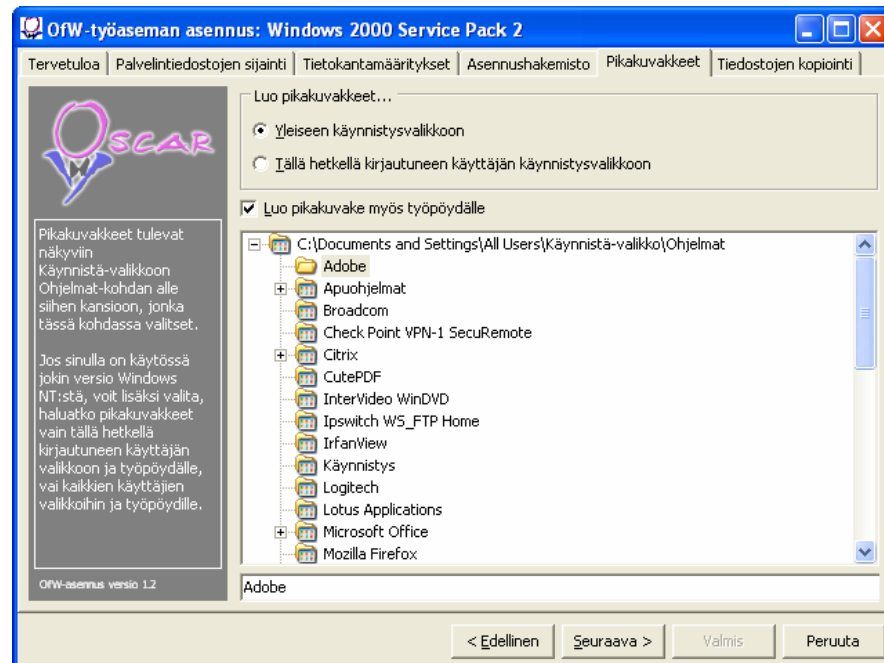
Kuva 3 OfW:n työaseman asennusohjelman tietokantamääritykset sisältävän tiedoston määrittämisruutu

4. Tässä vaiheessa määritetään asennushakemisto minne Oscar for Windows -ohjelma asennetaan kuvan 4 mukaisesti. Yleensä ohjelma asennetaan hakemiston "C:\Program Files" alle. Ohjelma luo itse määritettyyn hakemistoon oletuksena "Oscar for Windows"-hakemiston, jonka alle ohjelma lopulta asennetaan. Hakemiston määrittämisen jälkeen painetaan Seuraava-nappia asennuksen jatkamiseksi.



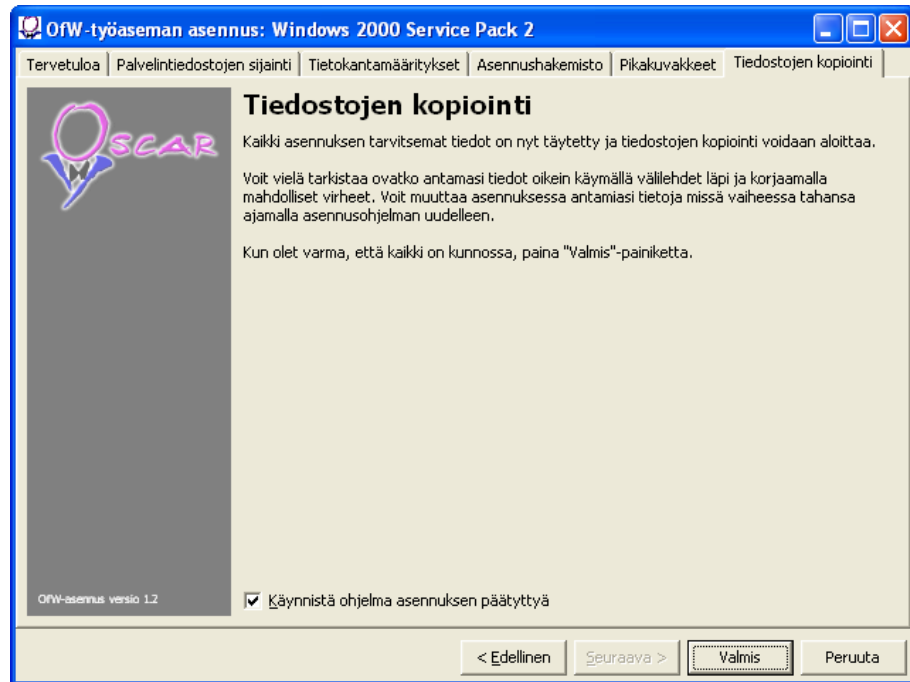
Kuva 4 OfW:n työaseman asennusohjelman asennushakemiston määrittämisruutu

5. Asennusohjelma kysyy käyttäjältä kuvan 5 mukaisesti miten menetellään pikakuvakkeiden asennuksen suhteen. Valintojen jälkeen painetaan Seuraava-nappia.



Kuva 5 OfW:n työaseman asennusohjelman pikakuvakkeiden määrittämisruutu

6. Lopuksi ohjelma ilmoittaa asetusten olevan valmiina asennusta varten kuvan 6 mukaisesti. Ruudulla on myös kohta ”Käynnistä ohjelma asennuksen päätyttyä”, jonka käyttäjä voi halutessaan valita. Asennus alkaa Aloita-nappia painamalla.



Kuva 6 OfW:n työaseman asennusohjelman yhteenvetoruutu

- Asennusohjelma kopioi tiedoston Ofwcheck.exe käyttäjän määrittämään asennushakemistoon. Samalla Windowsin rekisteriin kirjoitetaan muistiin tieto siitä, missä palvelintiedostot sijaitsevat. Kun ohjelma käynnistetään, käy tämä ohjelma tarkistamassa serveriltä hakemistosta ”Client” tiedoston Client.zop päivämäärän, ja mikäli kyseessä on ensimmäinen asennus tai tiedosto on uusi, niin se laajennetaan työaseman asennuksessa määritettyyn Oscar for Windowsin asennushakemistoon. Tiedosto Client.zop sisältää kaikki Oscar for Windowsin käytössä tarvittavat tiedostot. Jatkossa kun Oscar for Windowsista tulee uusi versio, suoritetaan palvelimen asennus, jolloin Client.zop tiedosto päivittyy. Aina kun Oscar for Windows työasemalla käynnistetään, se käy tarkistamassa palvelimelta Client.zop-tiedoston tilanteen ja tarvittaessa päivittää työasemalle uuden version Oscar for Windows -ohjelmasta laajentamalla palvelimelta uudemman Client.zop-tiedoston Oscar for Windowsin asennushakemistoon. Samalla palvelimen ”Client”-hakemiston alta löytyvästä hakemistosta ”User” kopioidaan sen sisältämät tiedostot työasemalle. Hakemistossa sijaitsee lisenssitiedosto, ja sinne voidaan jälkepäin lisätä myös muita tiedostoja, joita on tarkoitus siirtää kaikille työasemille.