



LAUREA
AMMATTIKORKEAKOULU

Uuden edellä

Sovelluspaketointi yritysympäristössä

Pystynen, Miikka

2011 Laurea Kerava

Laurea-ammattikorkeakoulu
Laurea Kerava

Sovelluspaketointi yritysympäristössä

Miikka Pystynen
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Elokuu, 2011

Pystynen, Miikka

Sovelluspaketointi yritysympäristössä

Vuosi 2011

Sivumäärä 39

Tämän opinnäytetyön tavoitteina oli tarkastella erilaisia sovelluspaketointimenetelmiä sekä tutkia, mitkä niistä olisivat sopivimpia erityisesti Laurea-ammattikorkeakoulun työympäristössä. Työssä keskityttiin tutkimaan erityisesti MSI-paketointia sekä erilaisia skriptausmenetelmiä. Työn toimeksiantajana toimi Laurean IT-palvelut. Työ jakautuu teoriaosuuteen ja käytännön osuuteen.

Työn teoriaosuudessa käsitellään sovelluspaketointia yleisesti. Siihen liittyen on käyty läpi sovelluspaketointiin liittyviä tekniikoita sekä apuvälineitä, kuten virtualisointia. Lisäksi teoriaosuudessa on listattuna paketointiohjelmiä.

Opinnäytetyön käytännön osuudessa perehdyttiin tarkemmin kahteen erilaiseen paketointiohjelmaan. Toinen ohjelma on MSI-paketointiohjelma EMCO MSI Packager ja toinen skriptauspohjainen WPKG. Näiden ohjelmien perustoiminnallisuuksia ja toimintaperiaatteita on tarkasteltu. Lisäksi käytännön osuudessa on kuvattuna, kuinka sovelluspakettien luominen kyseisillä ohjelmilla onnistuu käytännössä.

Työn testauksessa käytettiin apuvälineenä virtualisointia. Tässä tapauksessa Microsoftin Windows Server 2008R2 käyttöjärjestelmän sisältämän työaseman päällä toimivaa Microsoftin Hyper-V virtualisointiohjelmistoa. Kyseiseen virtuaaliympäristöön oli asennettu virtuaalikoneita, joiden avulla pystyttiin luomaan helposti hallittava testausympäristö ja nopeuttamaan testausprosessia.

Sovellusten paketoinnissa täytyy kiinnittää erityistä huomiota siihen, että paketteihin ei pääse mitään ylimääräistä tietoa. Tämä on ongelmana erityisesti MSI-paketoinnissa. Työn käytännön osuudessa tulikin lukuisia ongelmia erityisesti MSI-paketoinnin yhteydessä, vaikka kaikki tarpeelliset varotoimenpiteet oli tehty. MSI-paketit on helppo jakaa käyttäjien työasemille, mutta niiden toimintavarmuus on jättänyt itsestään epävarman kuvan.

Huomattavasti varmatoimisemmaksi paketointimenetelmäksi osoittautui skriptaustiedostoihin perustuva WPKG. Ohjelmalla luodut paketit toimivat käytännössä aina, jollei tiedostojen tekijä ole tehnyt kirjoitusvirheitä. Ohjelman ongelmana on kuitenkin jossain määrin hankalaho sovelluspakettien jakelu suurelle määrälle työasemia. Pienille koneryhmille tämä ohjelma osoittautui erinomaiseksi ratkaisuksi. Ohjelmalla on kuitenkin potentiaalia toimia myös laajamittaisen sovellusjakelun apuna.

Asiasanat: sovelluspaketointi, virtualisointi, sovellusjakelu, MSI

Pystynen, Miikka

Software Packaging in Business Environment

Year	2011	Pages	39
------	------	-------	----

The main goals of this Bachelor's thesis were to examine as many different software packaging methods as possible, and to investigate which ones would be most useful in the work environment of Laurea University of Applied Sciences. The main focus of the thesis was to investigate MSI packaging as well as different scripting methods. This thesis was commissioned by the IT services of Laurea University of Applied Sciences. The thesis is divided into theory and empirical sections.

In the theoretical part of this thesis software packaging is studied quite generally. Several packaging methods and tools have been researched, for example virtualization. In addition to that there are several packaging programs described and listed.

In the empirical part of the thesis the main purpose was to study two different packaging programs and familiarize the readers of this thesis to them. The first tested program is MSI based packaging program EMCO MSI Packager, and the second one a script-based program WPKG. The basic functionalities and working principles of these programs were studied with care. Also, the package creation process is described.

In the package testing process, virtualization was used as a tool. In this case, a workstation with Microsoft Windows Server 2008R2 operating system that had Microsoft Hyper-V virtualization software installed. Several virtual workstations were installed in the virtual environment. This process helped to create an easily manageable test environment and made the testing process a lot smoother.

When creating software packages, it should be made certain that no redundant information gets into the packages. This is a major problem in MSI packaging. Several problems arose during the testing, even though all the necessary precautions had been made. It is easy to distribute MSI packages to the workstations of users, but the packages seem to work a bit randomly. Therefore the use of these rarely working packages is not recommended.

The Script-based WPKG proved to be a significantly more reliable packaging method. The packages made with this program worked practically always, unless the files contained typing errors. The only major problem of this program is large-scale software distribution. WPKG proved to be an excellent program when used with a small specific user group. However, the program has even potential to work in large-scale software distribution, too, even though this requires some additional research on the subject.

Key Words: software packaging, virtualization, software distribution, MSI

Sisällys

Sanaluettelo	7
1 Johdanto	8
2 Windows Installer	8
2.1 Windows Installer 3.0	9
2.2 Windows Installer 4.5	9
2.3 Windows Installer 5.0	10
3 MSI-paketointi	10
3.1 MSI-paketoinnin menetelmät	10
3.1.1 Monitoring	11
3.1.2 Snapshot	11
3.1.3 MSI-pohjaiset asennusskriptit	11
4 Skriptausasennus	12
4.1 Skripti parametreilla	12
4.2 Skriptin asennustiedostot	12
5 Paketointityökaluja	13
5.1 WPKG	14
5.2 MSIBuilder	14
5.3 MakeMSI	14
5.4 MSIOrca	15
5.5 The AppDeploy Repackager	15
5.6 EMCO MSI Package Builder	15
6 Virtualisointi	16
6.1 Virtualisoinnin hyötyjä paketoinnissa	16
6.2 Microsoft Hyper-V	17
7 Opinnäytetyöprojektin toteutus	17
7.1 Projektin valmistelut	18
7.2 Virtuaalikoneiden asennuksessa ilmenneitä ongelmia	20
7.3 Projektin testaus	21
7.3.1 Riskianalyysi	21
7.3.2 Riskianalyysin osa-alueet	21
7.4 Asennuspaketin luominen WPKG-ohjelmistolla	23
7.4.1 WPKG:n käyttöönotto	23
7.4.2 Ajatuksia WPKG:n käytöstä Laureassa	28
7.5 Asennuspaketin luominen EMCO MSI Package Builderilla	29
7.5.1 EMCO MSI Package Builderin käyttöönotto	29
7.5.2 Ajatuksia EMCO MSI Builderin käytöstä Laureassa	34
8 Johtopäätökset	35

Lähteet	37
Kuvat ja kuviot	38
Taulukot	39

Sanaluettelo

Active Directory	Windowsin Aktiivihakemisto, voidaan käyttää esimerkiksi sovel-lusjakeluun.
Group Policy	Aktiivihakemiston Ryhmäkäytänteet.
Hyper-V	Microsoftin virtualisointialusta.
JS	Javascript-tiedostomuoto, jolla voidaan ajaa toisia tiedostoja.
Monitoring	MSI-pakettien luontiin käytettävä menetelmä, nauhoitus.
MSI	Microsoft Installer, Windows asennuspalvelun tiedostomuodot.
Msiexec.exe	Windows Installer käyttää Msiexeciä MSI-pakettien asennuksessa.
SCCM	System Center Configuration Manager, järjestelmänhallinnan so-vellus, voidaan käyttää esimerkiksi sovellusjakeluun.
Snapshot	MSI-pakettien luontiin käytettävä menetelmä, kuvaus.
Skripti	Komentosarjakieli, sen avulla pystytään automatisoimaan tehtä-viä ilman varsinaista ohjelmointikieltä.
Windows Installer	Windowsin asennus- ja kokoonpanopalvelu.
XML	Extensible Markup Language, rakenteellinen kuvauskieli.

1 Johdanto

Tämän opinnäytetyön tarkoituksena on toimia ohjeistuksena yrityksille, erityisesti Laurea-ammattikorkeakoululle. Työn tarkoituksena on tutkia eri sovelluspaketointimenetelmiä ja löytää niistä paras mahdollinen, jota voitaisiin hyödyntää Laurean työympäristössä. Työn teoriaosio käsittelee eri paketointimetojeja sekä niihin liittyviä tekniikoita, kun taas käytännön projektiosuudessa on testattu ja pohdittu kahta eri ohjelmaa, jotka voitaisiin ottaa käyttöön eri tarkoituksiin yleisesti Laureassa. Työn toimeksiantajana on toiminut Laurean IT-palvelut.

Suuren organisaation näkökulmasta työasemien ylläpidossa on tärkeää, että käyttöjärjestelmät sekä ohjelmat voidaan asentaa koko organisaation koneisiin tehokkaasti ja nopealla aikataululla. Mikäli organisaatiossa ilmenee tarvetta muuttaa jopa satoihin työasemiin asennettuja ohjelmistoja, työmäärä olisi aivan liian suuri, mikäli asennukset jouduttaisiin tekemään manuaalisesti. Manuaalisella asennuksella tarkoitetaan sitä, että ylläpitohenkilö käy yksitellen jokaisella työasemalla ja tekemään ohjelmistojen asennukset paikan päällä. Kyseinen operaatio vie runsaasti työtunteja ja rasittaa henkilöstöä. Täten tähän tarvitaan muita keinoja.

Manuaalisten asennusten minimoimiseksi onkin kehitetty erilaisia paketointimenetelmiä. Suosituimmat menetelmät ovat MSI-paketointi ja erilaisten asennuskriptien avulla tehdyt asennukset. Näiden menetelmien etuna on se, että sovellukset pystytään jakamaan etäasennuksina verkon yli käyttäjien työasemille niin, ettei ylläpitäjän tarvitse käydä itse työasemilla lainkaan. Lisäksi asennukset voidaan tehdä kokonaan niin sanottuina hiljaisina asennuksina, jolloin käyttäjälle ei koidu välttämättä minkäänlaista haittaa asennusoperaatiosta.

Valmiit sovelluspaketit voidaan asentaa käyttäjien koneille esimerkiksi käyttämällä Windowsin aktiivihakemistoa (engl. Active Directory). Tämä työ keskittyy kuitenkin nimenomaan sovelluspaketointiin sekä parhaan paketointimenetelmän löytämiseen, joita voidaan käyttää hyödyksi Laureassa. Aktiivihakemiston osuus on pyritty minimoimaan.

2 Windows Installer

Windows Installer on asennus- ja kokoonpanopalvelu, joka on käytössä Windows-ympäristössä. Asennuspalvelun luoma standardi mahdollistaa helpot sovellusten asennukset sekä yrityksille että yksityisille asiakkaille. Windows Installer mahdollistaa myös sovellusten sekä ominaisuuksien mainostamisen asentamatta niitä kuitenkaan koneelle. (Microsoft Corporation 2010.)

Windows Installer teknologia otettiin ensimmäisen kerran käyttöön kun Windows 2000 julkaistiin, tarkoituksena helpottaa sovellusten jakelua ja hallintaa yritys ympäristössä. Vanhemmissa

Windows-versioissa asennuspaketit luotiin yleensä kolmannen osapuolen suunnittelemissa työkaluilla. Pyrkimyksenä yhdenmukaistaa työkaluja, Microsoft lisäsi Windows Installerin (msiexec.exe) osaksi Windowsin ydinpalveluita (engl. core services). Tarkoituksena mahdollisuus asentaa, korjata ja poistaa ohjelmien asennuksia. Ohjeet asennukseen löytyvät MSI-tiedostoista, jotka ovat käytännössä tietokantatiedostoja, jotka sisältävät kaiken tiedon, jota sovelluspaketti tarvitsee asentuaikseen. Valmis sovelluspaketti voidaan jakaa käyttäjien työasemille ryhmäkäytäntöä käyttämällä (engl. Group Policy). (Tulloch 2005.)

Kun Microsoft oli julkaissut Windows Installerin, otti yritys toimintaperiaatteekseen lisätä MSI-asennuspaketit jokaiseen Windowsiin suunniteltuun sovellukseen. Tässä vaiheessa Microsoft ei kuitenkaan lisännyt käyttöjärjestelmäänsä työkalua, jolla voitaisiin uudelleenpaketoita perinteiset Setup-pohjaiset sovellukset MSI-paketeiksi. Yritys päättikin lopulta luoda WinINSTALL-ohjelmasta kevyemmän WinINSTALL LE:n, joka lisättiin Windows 2000:n asennuslevylle. Tämän sovelluksen avulla järjestelmänvalvojat pystyivät muuttamaan vanhat sovellukset MSI-paketeiksi, jotka pystyttäisiin jakamaan asiakkaiden työasemille ryhmäkäytäntöä käyttämällä. Microsoft päättikin jättää kolmannen osapuolen ohjelmistokehittäjille MSI-paketointityökalujen kehittämisen. (Tulloch 2005.)

Tämän valinnan seurauksena markkinoille alkoi ilmestyä lukuisia MSI-paketointityökaluja. Jotkut näistä ohjelmista ovat ilmaisia vapaaseen lähdekoodiin perustuvia ja toiset kaupallisia, jotka maksavat satoja euroja. Näistä sovelluksista löytyy lisää tietoa MSI-paketointia käsittelevissä luvuissa.

2.1 Windows Installer 3.0

Windows Installerin 3.0 -versiolla on mahdollista suorittaa monenlaisia toimintoja yhden asennusoperaation aikana, kuten asentamaan useita päivityksiä samaan aikaan.

Sovellusten päivitysjärjestystä voidaan muuttaa, jolloin asennusoperaatio sujuu nopeammin. Sama pätee myös päivitysten poistamiseen, eli päivitykset voidaan poistaa missä järjestyksessä tahansa. (Microsoft Corporation 2010.)

2.2 Windows Installer 4.5

Windows Installer 4.5 -version merkittävin uusi ominaisuus on kyky asentaa useita asennuspaketteja samanaikaisesti. Tämä onnistuu käyttämällä transaction processing-menetelmää. Tässä menetelmässä useampi prosessi nivoutuu yhteen muodostaen yhden kokonaisuuden. Mikäli jokin prosessin osa-alueista epäonnistuu, kokonaisprosessi palautuu alkuperäiseen muotoonsa.

Windows Installerin tapauksessa tällä varmistetaan se, että kaikki asennuspaketit varmasti asentuvat. Mikäli yhdenkin asennus epäonnistuu, kaikkien asennusohjelma palauttaa järjestelmän tilaan, jossa se oli ennen asennuksen alkamista. (Microsoft Corporation 2010.)

2.3 Windows Installer 5.0

Windows Installer 5.0 -versio mahdollistaa käyttäjäkohtaisten asetusten määrittämisen. Näihin lukeutuu esimerkiksi oikeuksien määrittäminen tietyille käyttäjille.

Asennuspaketti voidaan määrittää turvaamaan uusia käyttäjätilejä, Windowsin palveluita, tiedostoja ja kansioita. Näiden lisäksi myös rekisteriavaimien muokkaus onnistuu tämän version avulla (Microsoft Corporation 2010.)

3 MSI-paketointi

Ennen kuin sovelluksia voidaan alkaa paketoita, tulisi varmistaa, että asennuspakettien luontiin tarkoitettu työasema on puhdas. Tällä tarkoitetaan sitä, ettei kyseisellä koneella ole mitään ylimääräisiä ohjelmia, kuten virustorjuntaa, joka saattaa häiritä asennuspakettien luomisprosessia. Asennuspakettien tekemiseen tarkoitetut ohjelmat käytännössä nauhoittavat kaiken, mikä koneella muuttuu määritellyn ajan sisällä. Esimerkiksi toisten ohjelmien tekemät muutokset saattavat häiritä koko prosessia. Täten sovelluspaketointi onnistuu parhaiten käyttämällä tietokonetta, jonka käyttöjärjestelmä on puhdas kaikista ohjelmista. (Ruuska 2006, 13.)

Virtuaalinen käyttöjärjestelmä sopii erinomaisesti sovelluspaketointiin, sillä siihen tarvitaan nimenomaan käyttöjärjestelmä, joka on puhdas ylimääräisistä sovelluksista. Hyper-V virtualisointiohjelmistolla pystytään palauttamaan käyttöjärjestelmä nopeasti muutosta edeltävään tilaan, jolloin paketoinnin valmistukseen tarkoitettu käyttöjärjestelmä on helppo pitää puhtaana. Täytyy kuitenkin muistaa, ettei asiantuntija virtuaalikonetta sulkiessaan tallenna tehtyjä muutoksia käyttöjärjestelmään. (Juntunen 2007, 18.)

3.1 MSI-paketoinnin menetelmät

MSI-pakettien valmistamiseen käytetään yleisesti kahta eri menetelmää. Nämä ovat "Monitoring" ja "Snapshot". MSI-paketteja voidaan lisäksi valmistaa käyttämällä hyväksi erilaisia asennuskriptejä. Menetelmät ovat helppokäyttöisiä, mutta tutkittaessa asiaa tarkemmin selviää, että nämä menetelmät eivät välttämättä toimikaan riittävän varmasti. (Ruuska 2006, 6.)

Näiden niin sanottujen automatisoitujen menetelmien ongelmana onkin usein, että kaikkea ohjelman käyttäjän haluamaa tietoa ei saadakaan talteen. Toinen mahdollinen ongelmatilanne on, että menetelmien mukana tulee liikaa turhaa tietoa, joka saattaa vaarantaa koko paketin toimivuuden.

3.1.1 Monitoring

Tätä tekniikkaa käytettäessä ohjelma nauhoittaa jokaisen tapahtuman, jota sovelluksen asennusoperaatio työasemalle suorittaa. Nauhoitus ottaa huomioon myös esimerkiksi rekisteriin tapahtuvat muutokset. (EMCO 2011.)

Tällä menetelmällä voidaan esimerkiksi poistaa vanha versio ohjelmasta ja asentaa uusi versio tilalle ja luoda tästä koko operaatiosta MSI-paketti. Nauhoituksen avulla saatujen muutostietojen perusteella saadaan luotua MSI-asennuspaketti. (EMCO 2011.)

3.1.2 Snapshot

Snapshot menetelmää käytettäessä paketointiohjelma ottaa käytetystä työasemasta kaksi sisäistä kuvaa (engl. image). Nämä kuvat sisältävät käytännössä kaiken kovalevyllä sillä hetkellä tallennetun tiedon.

Ensimmäinen kuva otetaan ennen kuin paketoitava sovellus asennetaan ja toinen välittömästi asennuksen jälkeen. Näitä kuvia vertaamalla sovellus pystyy havaitsemaan työasemaan tapahtuneet muutokset ja luomaan niiden pohjalta MSI-paketin. (Ruuska 2006, 6.)

3.1.3 MSI-pohjaiset asennuskriptit

Niin sanottujen tavallisten MSI-paketointimetodien lisäksi tulee muistaa, että MSI-paketteja voidaan luoda myös asennuskriptejä hyväksi käyttäen. MSI-paketin luonti manuaalisesti on asiaan perehtymättömälle haastavaa, joten tätä tarkoitusta varten on luotu lukuisia erilaisia maksuttomia ja maksullisia ohjelmia.

Näiden ohjelmien tarkoituksena on siis avustaa IT-asiantuntijoita luomaan mahdollisimman helposti MSI-paketteja, jotka pystytään jakamaan helposti ja nopeasti käyttäjien työasemille. Esimerkkityökaluna mainittakoon WPKG, jonka avulla pystytään helposti asentamaan sovelluksia käyttäjien koneille muokkaamalla muutamaa asetustiedostoa. (WPKG.org 2010.)

4 Skriptausasennus

Kuten MSI-paketoinnissa, myös asennuskriptejä luotaessa ja testausprosessissa on hyvä käyttää virtuaaliympäristöä hyväksi. Käyttöjärjestelmän puhtaus ei ole yhtä kriittistä kuin edellä mainituissa MSI-pakettien luontimetodeissa. Sovellusten asennus voidaan suorittaa nauhoittamatta sovellusten toimintaa. Täten sovellusten asentaminen sujuu todennäköisesti ilman ongelmatilanteita, sillä toiset sovellukset eivät häiritse asennusta.

Asennuskriptien suurin etu MSI-paketteihin nähden onkin juuri toimintavarmuus. MSI-paketteihin saattaa eksyä helposti ylimääräistä tietoa, joka voi häiritä koko paketin toimintaa, ja virheen etsiminen saattaa olla työläs prosessi. Kun asennus tehdään skriptejä käyttämällä, ylimääräistä tietoa ei käytännössä voi esiintyä asennuksen yhteydessä.

4.1 Skripti parametreillä

Sovelluksen asentaminen voidaan tehdä antamalla oikeat parametrit suoritettavalle asennusohjelmalle. Eri parametreilla voidaan käytännössä saavuttaa täysin samat ominaisuudet kuin MSI-paketteja käyttämällä. Parametreja käyttäessä tulee kuitenkin olla tarkkana, että ohjelmalle annetaan oikeat komennot, jotta sovellus asentuu oikein.

Parametrien käyttäminen edellyttää tiettyä ammattitaitoa järjestelmän ylläpitäjältä. Skriptejä tehdessä tapahtuukin joskus inhimillisiä erehdyksiä, joiden seurauksena sovellukset eivät asennu oikein käyttäjien koneille.

4.2 Skriptin asennustiedostot

Asennuskripteissä voidaan siis käyttää hyväksi valmiita asennustiedostoja. Skriptaustiedostossa voidaan määritellä ominaisuuksia asennustiedostolle, käyttää useampaa asennustiedostoa tai vaikka luoda uusi asennuspaketti käyttämällä siihen soveltuvaa sovellusta.

Skriptausta auttamaan on kehitetty lukuisia sovelluksia, joiden avulla voidaan luoda erilaisia sovelluspaketteja. Erityisen tärkeää on kuitenkin pitää mielessä, että skriptiin määritellään tarkasti oikeat tiedostopolut, joissa asennettavat sovellukset sijaitsevat. Lisäksi määritellyt parametrit tulisi tarkistaa kahteen kertaan. Kirjoitusvirheitä tapahtuu kuitenkin aika ajoin. Todistettavasti yksikin väärä merkki asennuskriptissä pystyy aiheuttamaan koko organisaation työasemissa vakavia ongelmia.

5 Paketointityökaluja

Paketointiohjelman ja -menetelmän valintaan vaikuttaa moni eri asia. Ensinnäkin hyvin tärkeää on tietää, miten sovellukset tulisi jakaa käyttäjien työasemille. Sovellusjakeluun on käytössä useita eri tapoja, joihin ei tässä teoksessa erityisesti keskitytä. Aihe on yksinkertaisesti riittävän laaja, jotta siitä voisi helposti kirjoittaa oman opinnäytetyönsä. Varmasti yleisin sovelluspakettien jakelumetodi yrityksissä on aktiivihakemiston ryhmäkäytäntöjen (engl. Group Policy) kautta. Muitakin jakelumetodeja löytyy, kuten SCCM, mutta kuten mainittua, tässä työssä keskitytään sovelluspaketointiin, ei niinkään sovellusjakeluun.

Toinen vähintään yhtä tärkeä asia, joka tulee ottaa huomioon sovellusmenetelmää valittaessa, on yksinkertaisesti ohjelman sekä menetelmän käyttöönoton helppous. Tutkittaessa eri paketointiohjelmaa huomaa nopeasti, kuinka hankalia monien ohjelmien käyttö on. Muutamien paketointimenetelmien käyttöönotto tarvitsee jopa oman, pelkästään tiettyä ohjelmaa varten tehdyn skriptauskielen opetteluun. Tämä toki mahdollistaa käytännössä täysin rajattomat mahdollisuudet ohjelman käyttöön, mutta ottaen resurssit huomioon, saattaa ohjelman käyttöönottoon kulua yksinkertaisesti liikaa aikaa.

Paketointimetodia valittaessa tuleekin ottaa yrityksen resurssit huomioon. Monet, varsinkin MSI-paketointisovellukset, ovat helppokäyttöisiä, mutta hyvän käytettävyyden ja helpon käyttöönoton vastapainona on usein kova hinta. Yrityksen tulisi miettiä tarkkaan, kuinka paljon se on valmis panostamaan sovelluspaketointiin käytettävään ohjelmaan.

Asennusskriptejä käytettäessä ei välttämättä tarvita mitään apuohjelmia, mutta monimutkaisten skriptien laatiminen ja erityisesti päivittäminen saattaa olla haastavaa. Yrityksen IT-osaston ammattitaidolla onkin merkittävä rooli siinä, kun mietitään millaisia sovelluksia yrityksessä voidaan ja kannattaa käyttää.

Monet IT-alan ammattilaiset osaavat käyttää asennusskriptejä luontevasti ja riittävän tehokkaasti. Kuitenkin saattaa olla järkevää miettiä joitain ohjelmia, jotka avustavat skriptien tekemisessä ja täten tehostavat työntekoa.

Seuraavissa luvuissa on kuvattuna useita erilaisia paketointisovelluksia, joita voidaan käyttää hyödyksi yritysympäristössä. Kuvaukset ovat tarkoituksella lyhyitä, sillä työn projektiosuudessa on testattu vain kahta sovellusta käytännössä. Täten rajallisen ajankäytön vuoksi ei ole järkevää testata kaikkia mahdollisia aiheeseen liittyviä sovelluksia.

5.1 WPKG

WPKG on automatisoitu ohjelma sovellusten paketointiin, jakeluun ja poistamiseen suunniteltu ohjelma. Se on suunniteltu Windows-käyttöjärjestelmälle. WPKG:ta voidaan käyttää hyväksi, kun IT-asiantuntijoiden täytyy esimerkiksi asentaa tai päivittää useita ohjelmia samanaikaisesti monelle eri työasemalle palvelimelta. Kyseinen operaatio onnistuu esimerkiksi Active Directoryn kautta. WPKG perustuu avoimeen lähdekoodiin ja on siten täysin ilmainen. (WPKG.org 2010.)

WPKG on varsin tehokas apuväline, sillä sitä käyttämällä voidaan asentaa MSI, InstallShield, PackagefortheWeb, Inno Setup ja Nullsoft asennuspaketteja. Näiden lisäksi exe, bat sekä cmd skriptiasennukset onnistuvat. Käytännössä tämän sovelluksen avulla pystyy siis suorittamaan kaikki nykyiset asennustavat sekä metodit. WPKG:lla voidaan asentaa sovelluksia niin sanottuna tausta-asennuksena (hiljainen asennus), jolloin asennus suoritetaan käyttäjän tietämättä taustaprosessina. (WPKG.org 2010.)

5.2 MSIBuilder

MSIBuilder on ilmainen Windows Installer -asennuspakettien luontiohjelma. MSIBuilderilla pystytään muokkaamaan sekä luomaan MSI-paketteja. Paketin muokkauksen perusominaisuudet, kuten rekisterimerkintöjen lisäys ja poisto onnistuvat ohjelmalla. (Brothersoft, 2008)

Ohjelmassa kiteytyy tietynlainen yleinen IT-alan sovellusten ongelma. Nimittäin ohjelmalla pystytään tekemään käytännössä mitä tahansa, mutta se on todella sekavan oloinen. Ohjelman käyttö vaatii siis runsaasti paneutumista, joten ohjelman käyttöönottoa tulisi miettiä tarkasti ja mahdollisesti kokeilla sitä hetken, ennen kuin sitä voitaisiin edes harkita käytettäväksi yritysympäristössä. (Brothersoft 2008.)

5.3 MakeMSI

MakeMSI on ilmainen MSI-pakettien luontiin tarkoitettu ohjelma. Ohjelma hyödyntää XML-tiedostoja, joilla pystytään helposti luomaan tai päivittämään MSI-asennuspaketteja. Ohjelma on helppokäyttöinen eikä vaadi juurikaan asiantuntemusta. (Bareis 2011.)

MakeMSI:llä voidaan joko luoda uusia MSI-paketteja tai muokata vanhoja. MakeMSI:lle voidaan osoittaa, mistä tietyn ohjelman hakemisto tai rekisterimerkintä löytyy, jolloin ohjelma osaa automaattisesti tehdä kaikki Windows Installerin vaatimat merkinnät.

Helpottaakseen virheiden paikannusta ja kertoakseen aloittelevalle käyttäjälle, mitä paketin luomisprosessin aikana tapahtuu, luo ohjelma html-muotoisen raportin koko prosessista. Raportista ilmenee esimerkiksi asennuspaketissa käytetyt tiedostot ja niiden käyttämät ominaisuudet. (Bareis 2011.)

5.4 MSIOrca

MSIOrca on ilmainen MSI-pakettien muokkausohjelma. Ohjelma sisältää graafisen käyttöliittymän, joka selkeyttää pakettien muokkausoperaatiota merkittävästi. Asennuspaketin ominaisuudet ilmestyvät kätevästi taulukoihin, joiden rivejä ja ominaisuuksia voidaan muokata nopeasti. (Microsoftin tuotetuki 2010.)

MSIOrcaa voidaan käyttää yhdessä jonkin toisen asennuspakettien luontiohjelman kanssa, jolloin saadaan ilmainen kokonaisuus, jolla pystytään tekemään samat asiat kuin kalliilla ammattilaiskäyttöön suunnatuilla ohjelmilla. Tämä taas vaatii jonkin verran ammattitaitoa ja aikaa, sillä molempien ohjelmien käyttö tulee opetella ja niitä tulee osata käyttää yhdessä. (Microsoftin tuotetuki 2010.)

5.5 The AppDeploy Repackager

AppDeploy Repackager on ilmainen paketointiohjelma. Tämä ohjelma kykenee ainoastaan luomaan MSI-paketteja, ei muokkaamaan niitä. Tämän sovelluksen paketointiominaisuus toimii Snapshot periaatteella, eli koneesta otetaan järjestelmäkuva ennen ja jälkeen sovelluksen asennuksen. Tämän jälkeen sovellus osaa tutkia muuttuneet tiedot ja muodostaa MSI-paketin. Ennen paketin luomista käyttäjä pystyy muokkaamaan paketin tietoja haluamikseen, kuten esimerkiksi etsimään puuttuvia rekisterimerkintöjä. (AppDeploy 2011.)

Appdeployn Repackager vaikuttaa suhteellisen helppokäyttöiseltä ohjelmalta. Ohjelman heikkouksiin kuuluu se, että sillä luotuja paketteja ei pystytä enää muokkaamaan tällä ohjelmalla, vaan siihen operaatioon tarvitaan jokin toinen sovellus. Kuitenkin mikäli paketteja ei tarvitse muokata jälkikäteen, ohjelma voisi olla kokeilemisen arvoinen. (AppDeploy 2011.)

5.6 EMCO MSI Package Builder

EMCO MSI Package Builder on maksullinen MSI-paketointiohjelma. Ohjelma on helppokäyttöinen ja sen käyttöönotto sujuu nopeasti ilman juuri minkäänlaista asiantuntemusta. Tämän mahdollistaa ohjelman intuitiivinen graafinen käyttöliittymä, jonka ansiosta asennuspaketin luominen onnistuu muutamassa minuutissa. (EMCO 2011.)

Ohjelma käyttää monitoring nauhoitusmenetelmää, jonka avulla pystytään luomaan nopeasti luotettavia asennuspaketteja. Tämä prosessi on täysin automaattinen, ohjelmalle täytyy käytännössä vain näyttää mitä sen tulee nauhoittaa. Esimerkiksi exe-tiedoston muuntaminen MSI-paketiksi käy vaivattomasti tämän menetelmän avulla. Samalla asennuspakettia on helppo muuttaa mieleisekseen. Valmistuneen MSI-paketin asennus työasemille onnistuu vaivattomasti Windowsin ryhmäkäytäntöjen avulla. (EMCO 2011.)

6 Virtualisointi

Virtualisoinnilla tarkoitetaan yleisesti asioita, joita ei konkreettisesti ole olemassa, mutta joita voidaan käyttää fyysisten vastineidensa tavoin. Virtuaalisia tietokoneita voidaan käyttää hyödyksi monella eri tapaa IT-alalla, mutta erityisesti sovelluspaketoinnin yhteydessä testauksen sekä jakelun helpottamiseksi virtuaalikoneiden käyttö onkin järkevää. (Juntunen 2007, 9.)

6.1 Virtualisoinnin hyötyjä paketoinnissa

Sovelluspaketointia toteutettaessa ja valmistuneita paketteja testattaessa on erittäin tärkeää, että pystytään luomaan testiympäristö, joka ei ole suorassa yhteydessä käytännön ympäristöön. Näin pystytään varmistamaan se, että testausympäristö ja käytännön järjestelmät eivät pysty suoranaisesti häiritsemään ja täten mahdollisesti sekoittamaan toisiaan. (Juntunen 2007, 9.)

Virtualisoidun testiympäristön hyötyihin voidaan lukea muun muassa riippumattomuus käytetystä raudasta. Virtuaalisia käyttöjärjestelmiä voidaan ajaa useita samalla tietokoneella, joten yksittäisen koneen tehokkuutta pystytään nostamaan merkittävästi. Useat samaan aikaan ajettavat uudet käyttöjärjestelmät vievät kuitenkin merkittävän määrän keskusmuistia, joten tietokoneen tulisi olla ajan tasalla. (Juntunen 2007, 9.)

Virtualisoinnista on lisäksi muitakin hyviä puolia. Erityisen kätevä ominaisuus on käyttöjärjestelmän sen hetkisen tilan tallentaminen, kuten myös käyttöjärjestelmän palautusmahdollisuus aloitustilaansa. Tämä on erittäin tärkeää paketointia testattaessa, sillä muuten aikaa kuluisi valtavia määriä, mikäli jouduttaisiin joka testauksen välillä puhdistamaan käyttöjärjestelmä manuaalisesti. Virtualisointiin on olemassa useita erilaisia ratkaisuja, joiden käyttötarkoitus vaihtelee hieman. (Juntunen 2007, 10.)

Virtuaaliohjelmistoista mainittakoon esimerkiksi VMware sekä Virtual PC. Tässä opinnäytetyössä päädyttiin käyttämään Windows Server 2008 R2:een valmiiksi integroitua Microsoftin Hyper-V virtualisointiohjelmistoa. Tähän ratkaisuun päädyttiin pitkälti ohjelman käytettävyyden sekä aikaisempien positiivisten käyttökokemusten perusteella. Kuten jo mainittu, ohjel-

misto on integroitu valmiiksi käyttöjärjestelmään, joten sen käyttö ei aiheuta lisäkustannuksia. (Juntunen 2007, 10.)

6.2 Microsoft Hyper-V

Virtualisoinnilla on nykyään merkittävä rooli yrityksissä. Virtualisointi tehostaa merkittävästi organisaatioiden tehokkuutta ja lisäksi sillä pystytään laskemaan merkittävästi sähkönkulutusta, joka on taas ympäristön kannalta erittäin hyvä asia. (Microsoft Corporation, 2009)

Windows Server 2008 R2 pystyy hyödyntämään Hyper-V virtualisointiohjelmistoa. Täten pystytään luomaan sekä virtuaalisia työasemia että palvelimia. Hyper-V antaa fyysisen tietokoneen resursseja virtuaalisen tietokoneen käyttöön, jolloin virtuaalista konetta voidaan käyttää aivan kuten oikeaa fyysistä konetta. Hyper-V:tä käytetään usein virtuaalisten palvelimien luomiseen, mutta kun Hyper-V:tä käytetään virtuaalisen työpöytäinfrastruktuurin (engl. Virtual Desktop Infrastructure) kanssa, Hyper-V:tä voidaan käyttää tehokkaasti myös tavallisten työasemien virtualisointiin. (Microsoft Corporation 2009.)

Hyper-V on helppokäyttöinen sen graafisen käyttöliittymän ansiosta. Virtuaalisia käyttöjärjestelmiä voi olla useita käynnissä samaan aikaan sekä niiden välillä liikkuminen on nopeaa ja helppoa. Virtuaaliset käyttöjärjestelmät pystytään palauttamaan alkutilaansa nopeasti, joka helpottaa ja erityisesti nopeuttaa sovelluspakettien testausprosessia.

7 Opinnäytetyöprojektin toteutus

Tämän opinnäytetyön projektin tarkoituksena on selvittää kahden erilaisen paketointiohjelman käyttökelpoisuutta yritys ympäristössä. Päätaavoitteena on selvittää, mikä pakkausmetodi ja sovellus sopii parhaiten Laurean IT-palveluiden käytettäväksi. Tarkoituksena on siis selvittää, miten pystyttäisiin helpottamaan organisaatiossa käytettävien sovellusten asennuksia ja niiden mahdollista ylläpitoa.

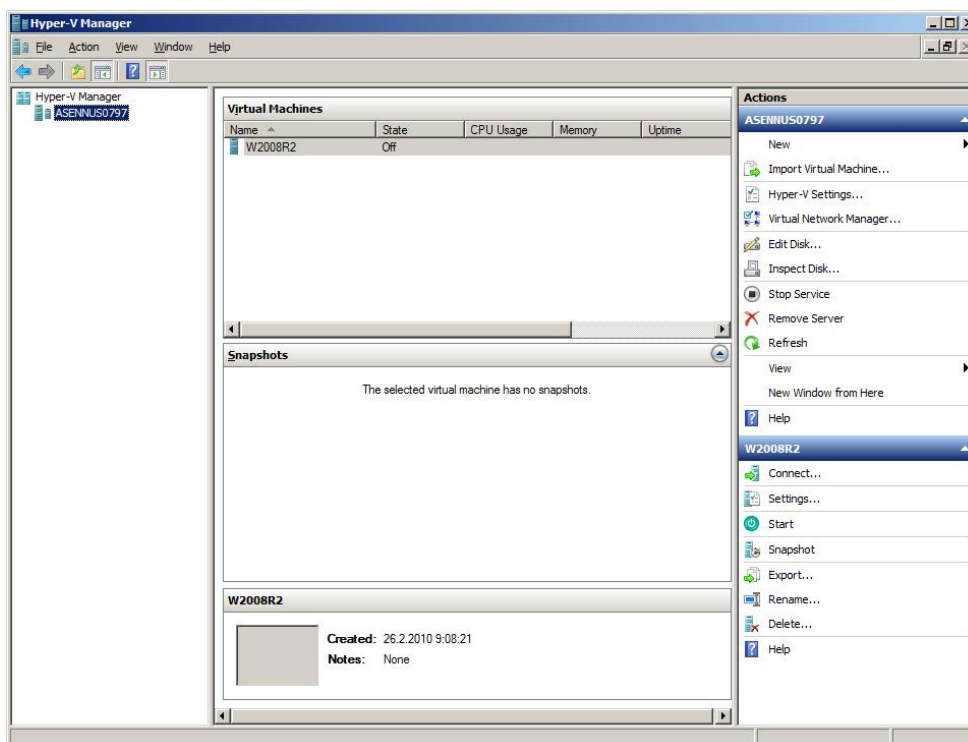
Jotta Laureassa pystyttäisiin hyödyntämään kyseisiä paketointimenetelmiä, tulee ohjelmien peruskäyttö opetella. Tietysti täytyy myös tutkia, miten paketointi valituilla paketointiohjelmilla toimii. Riittävän laajan yleiskuvan saamiseksi paketointiohjelmista tulee paketointia testata muutamalla testisovelluksilla, jotka ovat yleisesti käytössä eri yrityksissä. Testauksessa syntyvien pakettien toimivuus tulee testata, jotta käytettävien pakkausmenetelmien sekä ohjelmien mielekkyyttä voidaan arvioida. Mahdollisten virhetilanteiden sattuesssa tulisi pystyä selvittämään virheiden syyt ja seuraukset, haittaako virhe olennaisesti asennusta vai onko se esimerkiksi vain kosmeettinen. Lisäksi virhetilanteiden ehkäisyä tulisi pohtia, jotta pystyttäisiin tulevaisuudessa välttämään kyseisiä virhetilanteita.

Sovelluspaketointi opinnäytetyön aiheena on annettu tehtäväkseni, jotta Laurean IT-palvelut saisi merkittävää lisätietoa sekä eri paketointiohjelmistojen käytöstä että paketoinnista yleisesti. Opinnäytetyön tekijä puolestaan saa kehitettyä omaa tietotaitoaan tutustumalla täysin uuteen aihepiiriin ja kehitettyä tätä osaamistaan, josta on runsaasti hyötyä tulevaisuudessa.

7.1 Projektin valmistelut

Tämän projektin valmisteluihin kuului Microsoft Hyper-V -virtualisointiympäristön asennus Windows Server 2008R2 käyttöjärjestelmän omaavaan tietokoneeseen. Hyper-V:n avulla pystyttiin asentamaan useita virtuaalisia käyttöjärjestelmiä samalle tietokoneelle. Virtuaalikoneiden käyttö mahdollistaa puhtaiden käyttöjärjestelmien käytön, joiden avulla sovelluspaketointi onnistuu huomattavasti suuremmalla todennäköisyydellä kuin käyttöjärjestelmissä, joihin on asennettu valtavasti erilaisia ohjelmia. Tässä työssä ei perehdytä virtualisointiin tarkemmin, sillä aiheen laajuuden takia siitä saisi tehtyä melko vaivattomasti oman opinnäytetyönsä.

Hyper-V Managerilla pystytään hallitsemaan helposti virtuaalisia työasemia. Uusien työasemien lisäys, poisto ja muokkaus tapahtuu nopeasti ja käytännöllisesti. Kuvassa 1 näkyvät käytössä olevat virtuaalikoneet Virtual Machines kohdassa. Projektia varten asennettiin muutama virtuaalikone, joiden asennus onnistuu kohdasta New ja New Virtual Machine.

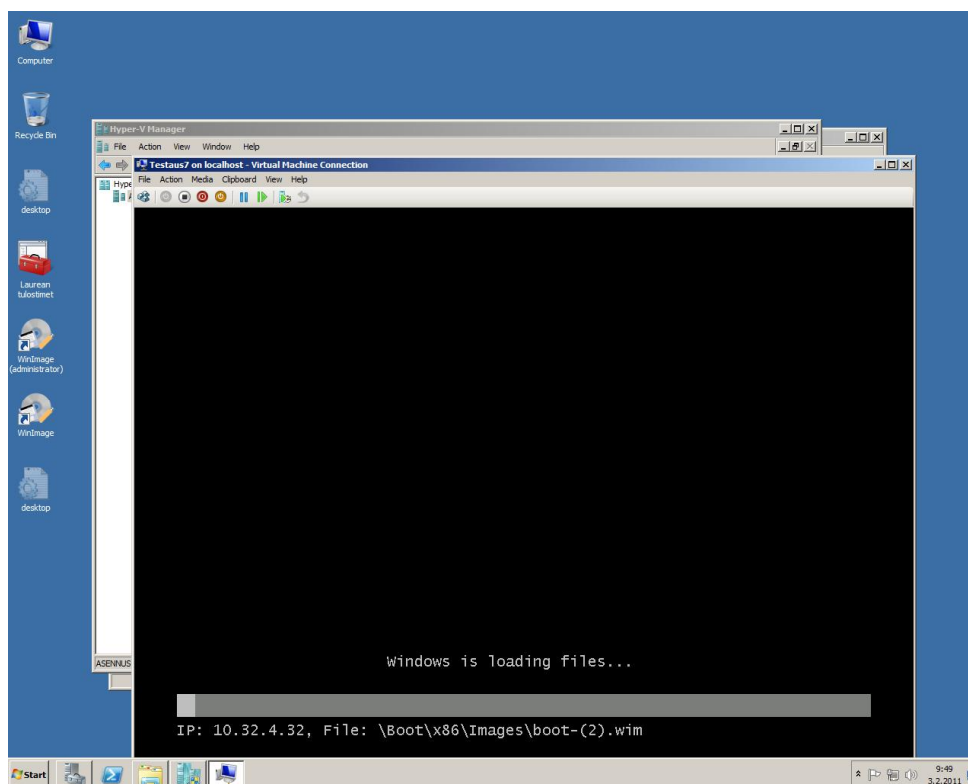


Kuva 1. Hyper-V Managerin pääikkuna.



Kuva 2. Uuden virtuaalikoneen määritykset valmiina.

Uuden virtuaalikoneen luominen on helppoa virtuaalikoneen asennusvelhon avulla (engl. Virtual Machine Wizard). Asennusohjelmassa on viisi kohtaa, joihin määritellään halutut asetukset. Kuvassa 2 asetukset on määriteltyinä. Tässä tapauksessa koneen nimeksi on annettu Paketointi. Seuraavassa kohdassa määritellään, kuinka paljon keskusmuistia annetaan fyysiseltä tietokoneelta virtuaalikoneen käyttöön. Tässä tapauksessa on valittu 1024 megatavua. Seuraavassa kohdassa valitaan koneen käyttämä verkkoyhteys, tässä valittuna Laurean Ulkoverkko. Seuraavaksi valitaan järjestelmäkuvan tiedoston asennuspaikka, eli missä virtuaalikone käytännössä sijaitsee kovalevyllä. Lisäksi voidaan valita virtuaalikoneen käyttämä kovalevytilan maksimimäärä, tällä virtuaalikoneella rajaton, eli sama kuin fyysisellä koneella. Lopuksi valitaan virtuaalikoneen käyttöjärjestelmän asennustapa, eli mistä käyttöjärjestelmä asennetaan virtuaalikoneelle kun se ensimmäisen kerran käynnistetään. Finish-painiketta painamalla uusi virtuaalikone luodaan.



Kuva 3. Windowsin asennus.

Kun virtuaalikone on asennettu, voidaan se käynnistää painamalla Start virtual machine -painiketta. Tämän jälkeen virtuaalikone käynnistyy aivan kuten perinteinen tietokone, paitsi omassa ikkunassaan toisen koneen sisällä. Seuraavaksi asennetaan käyttöjärjestelmä. Kuvassa 3 asentuu 64-bittinen Windows 7 verkkoyhteyden välityksellä palvelimelta. Kun käyttöjärjestelmä on asennettuna ja mahdollisesti päivitettyä, voidaan sitä käyttää aivan kuten tavallista tietokonetta.

Tässä luvussa oli kuvattuna lyhyesti Hyper-V virtuaalikoneiden asennusoperaatio pääpiirteissään. Tämä projekti ei keskity tämän tarkemmin virtualisointiin, koska se on vain työväline, jonka avulla helpotetaan sovelluspaketoinnin toimivuuden testausta.

7.2 Virtuaalikoneiden asennuksessa ilmenneitä ongelmia

Ensimmäiset projektissa ilmenneet ongelmatilanteet tulivat vastaan jo lähes alkumetreillä. Virtuaalikoneita asennettaessa palvelinkoneelle tuli vastaan itse koneen vanhentuneet komponentit. Vähäisen muistimäärän lisäksi tietokoneen kovalevy osoittautui aivan liian pieneksi.

Kun virtuaalikoneet oli jo asennettu, tehtiin päätös, että niitä ei enää aleta asentamaan uudestaan toiselle koneelle. Tämä johtikin siihen, että tuli keksiä jokin tapa, jolla saadaan ky-

seisen koneen tiedot kopioitua toiselle koneelle. Tässä päädyttiin sellaiseen ratkaisuun, jossa koneen kovalevy kokonaisuudessaan kopioitiin eli kloonattiin isommalle kovalevylle.

Tämä operaatio onnistui Symantec Ghost -ohjelman avulla, jolla muutenkin Laureassa asennetaan sekä myös kloonataan työasemia. Ongelmat eivät kuitenkaan loppuneet siihen. Tietokoneen tiedot saatiin siirrettyä onnistuneesti isommalle kovalevylle ja aluksi kaikki näytti toimivan hyvin.

Kuitenkin pian huomattiin, että jostain syystä virtuaalikoneet eivät enää toimi. Usean tunnin tutkimisen jälkeen selvisikin, että kloonausprosessi oli aiheuttanut sen, että Hyper-V -virtualisointisovelluksen vaatima eräs palvelu ei enää käynnistynytkään Windowsin mukana automaattisesti.

Pian huomattiin, että palvelua ei saanutkaan enää käynnistettyä automaattisesti käyttöjärjestelmän käynnistyessä Windowsin hallintatyökalujen kautta. Ongelma ratkesi vasta, kun kommentoriviltä palvelu oli manuaalisesti määritelty käynnistymään automaattisesti Windowsin käynnistyessä.

7.3 Projektin testaus

Tämän opinnäytetyöprojektin toiminnallisessa osuudessa pyritään testaamaan käyttötarkoitukseltaan sopivimmat paketointityökalut käytettäväksi yritysympäristössä. Tässä on otettu erityisesti huomioon hyvä käytettävyys, resurssit ja käyttöönoton helppous.

Projektin toiminnallisessa osuudessa on testattu kaksi erityyppistä paketointiohjelmistoa, jotka tarjoavat erilaisen lähestymistavan käsiteltävään aiheeseen. Seuraavassa luvussa testattuja ohjelmia on analysoitu riskianalyysin muodossa.

7.3.1 Riskianalyysi

Riskianalyysin tavoitteena on selvittää kahden eri paketointisovelluksen käyttöönotossa esiintyviä ongelmatilanteita, jotka saattavat vaikuttaa ohjelmiston käyttöönottoon. Tässä on pyritty tarkastelemaan erityisesti paketoinnin onnistumistodennäköisyyttä, sovellusjakelun helppoutta ja toimivuutta sekä sovellusten monipuolisuutta. (Teknologian tutkimuskeskus VTT 2011.)

7.3.2 Riskianalyysin osa-alueet

Kun paketointiprosessin eri osa-alueet on jaettu pieniin osiin, riskialueiden tarkastelu on merkittävästi helpompaa. Taulukosta 1 ilmenee testattujen sovellusten pärjääminen kriittisillä osa-alueilla. Kaikki osa-alueet on arvioitu asteikolla 1-5, lukuun ottamatta käytettävyyttä ja hintaa, jotka eivät ole yhtä painavia kriteerejä kuin muut taulukossa olevat osa-alueet ja ovat täten pisteytetty asteikolla 1-3.

Sovellus	Monipuolisuus (eri paketointimetodit yms.)	Käytettävyys (1-3)	Käyttönoton helppous	Paketoinnin onnistuminen	Sovellusjakelun monipuolisuus ja helppous	Hinta (1-3)
WPKG	4	1	3	4	3	3
EMCO	2	3	5	1	3	1

Taulukko 1. Riskianalyysin osa-alueet pisteytettynä.

Monipuolisuudessa WPKG pärjää paremmin kuin pelkästään MSI-paketteihin keskittyvä EMCO. Lukuisat erilaiset asennuskriptit antavat huomattavasti enemmän vaihtoehtoja ja mahdollisuuksia kuin pelkät MSI-paketit. Lisäksi WPKG:n internetsivustolta löytyy merkittävästi valmiiksi tehtyjä skriptejä, joita voidaan käyttää hyödyksi, vaikka ohjelmaa ei käytettäisikään.

Käytettävydessä EMCO on yksinkertaisesti parempi. Ohjelma on todella helppokäyttöinen kun sitä verrataan toiseen analysoitavaan tuotteeseen. Käyttönoton helppoudessa EMCO on myös huomattavasti parempi. Ohjelman ulkoasu antaa pohjan hyvälle käytettävyydelle. Lisäksi ohjelman asennus onnistuu keneltä tahansa ohjelmistoja ennen asentaneelta. WPKG:n asennusoperaatio vaatii hieman asiantuntijuutta, mutta ohjelman internetsivustolta (WPKG.org) löytyy hyvät ohjeet ohjelmiston asentamiselle ja käyttämiselle. Kuitenkin EMCO:n käyttönotto on merkittävästi helpompaa kuin toisen testattavana olleen sovelluksen asentaminen. (WPKG.org 2010.)

Paketoinnin onnistumisella tarkoitetaan sitä, kuinka varmatoimisesti ohjelmistolla pystytään luomaan sovelluspaketti. Tässä WPKG selviytyy huomattavasti paremmin. Ilman käyttäjän tekemiä virheitä sovellus toimii käytännössä juuri niin kuin sen pitääkin. EMCO osoittautui todella epävarmaksi tässä suhteessa. Valmiiden toimivien pakettien luominen osoittautui hankalaksi tällä ohjelmalla. Tämä heijastuu myös suoraan sovellusjakeluun. Mikäli paketit eivät toimi, on niitä turhaa yrittää jakaa loppukäyttäjien koneille. Kuitenkin paketointioperaation onnistuttua on MSI-pakettien jakelu käyttäjien koneille vaivatonta. WPKG:n sovellusjakelun helppous ei aivan pärjää MSI-paketeille, mutta pakettien varmatoimisuuden puolesta se aina-

kin onnistuu. Hinnassa WPKG päihittää kilpailijansa, sillä se on ilmainen ohjelmisto. EMCO MSI Builder maksaa noin 600 dollaria, joka on jo merkittävä summa rahaa. Kuten taulukosta 1 selviää, WPKG voittaa tämän arvion perusteella EMCOn ohjelman yhteispistein 18-15.

Lyhyenä yhteenvetona WPKG on varmatoiminen, luotettava, monipuolinen ja ilmainen paketoitiohjelmisto. WPKG:n käytössä ei havaittu merkittäviä riskejä, varsinkaan kun sen käyttöönotto on suhteellisen helppoa. Lisäksi se on ilmainen ohjelma.

EMCO MSI Package Builder on helppokäyttöinen sekä selkeä ohjelmisto. Sen asentaminen sekä käyttö on helppoa. Automatisoidut ominaisuudet aiheuttavat kuitenkin ongelmia. Sovelluksella tehdyt paketit toimivat hyvin harvoin, ja kun ne toimivat, eivät ne toimi aina oikein. Lisäksi ohjelmisto on suhteellisen kallis.

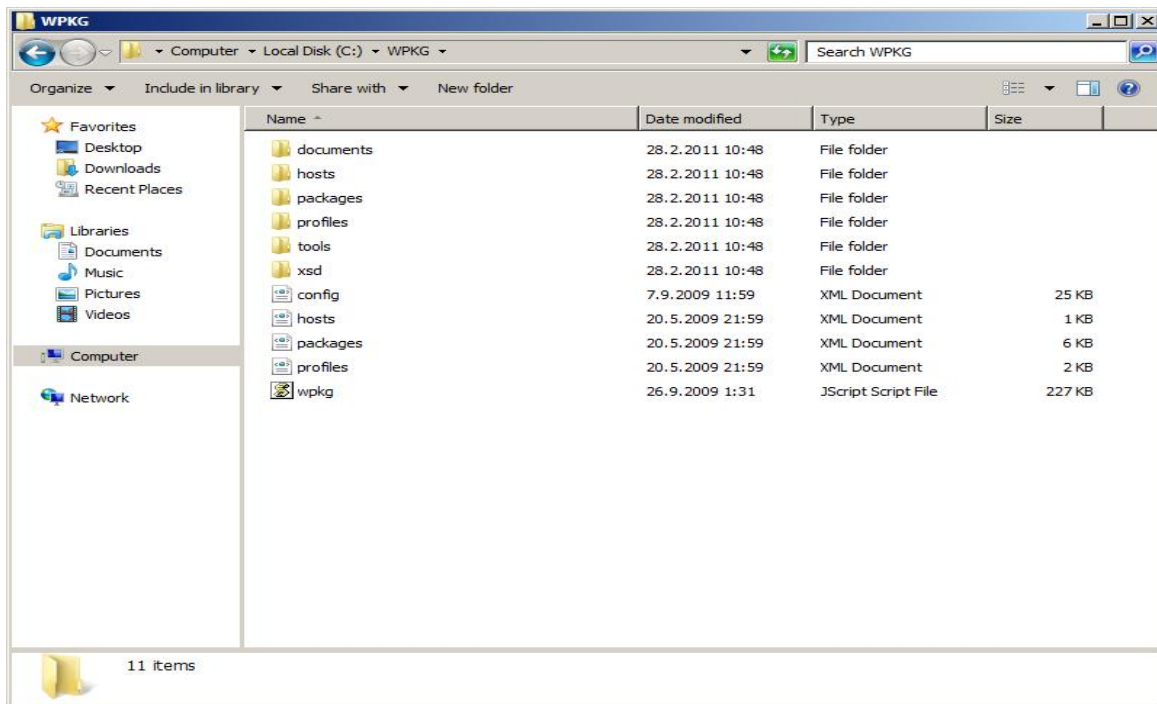
7.4 Asennuspaketin luominen WPKG-ohjelmistolla

WPKG on valittu projektissa käytettäväksi työkaluksi, sillä sen käyttöönotto on riittävän yksinkertaista. Tämä on ehkä painavin kriteeri ohjelman käyttöönoton kannalta, mikäli yrityksen resurssit ovat rajalliset.

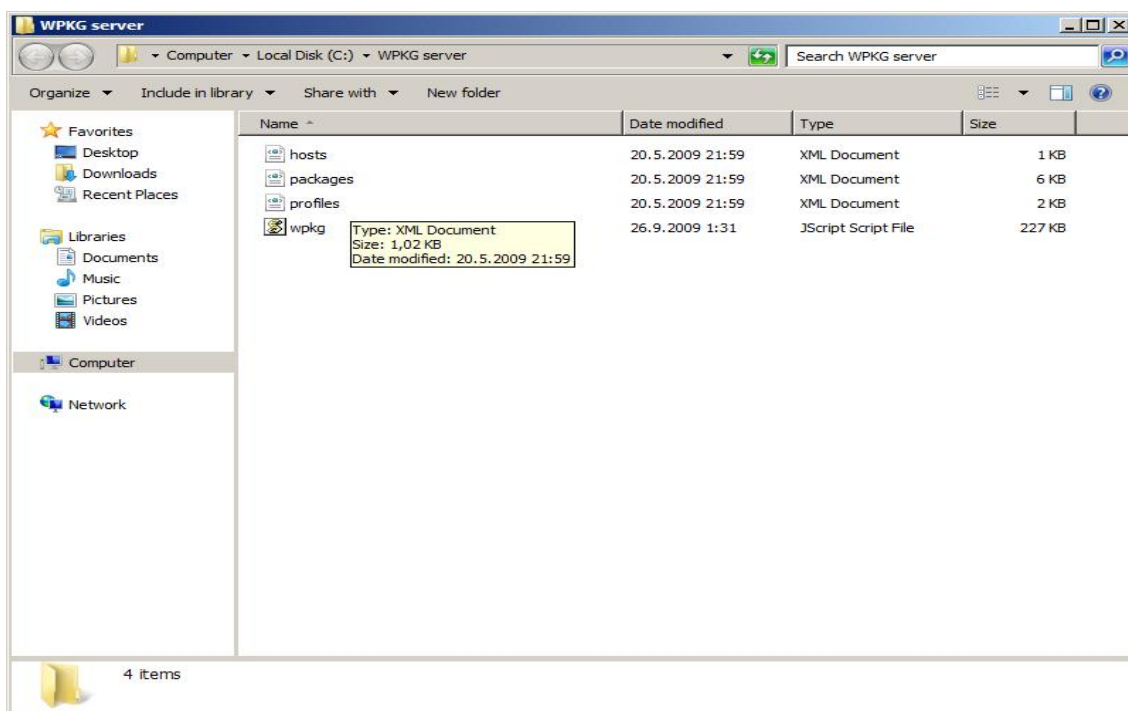
Pitäen resurssit mielessä ohjelma on myös täysin ilmainen käyttää, joka on vain hyvä asia. Vaikka ohjelma ei olekaan käytettävyydeltään välttämättä aivan parhaasta päästä, WPKG:n sivustolta (WPKG.org) löytyy hyvät dokumentaatiot, joiden avulla ohjelmaa oppii käyttämään nopeasti.

7.4.1 WPKG:n käyttöönotto

Jotta ohjelmaa pystytään käyttämään, tulee se luonnollisesti asentaa. WPKG:n asennus sujuu seuraavalla tavalla: WPKG:n sivuilta ladataan pakattu tiedostokansio, joka puretaan tietokoneen kovalevylle. Kuvassa 4 on kansio purettuna.



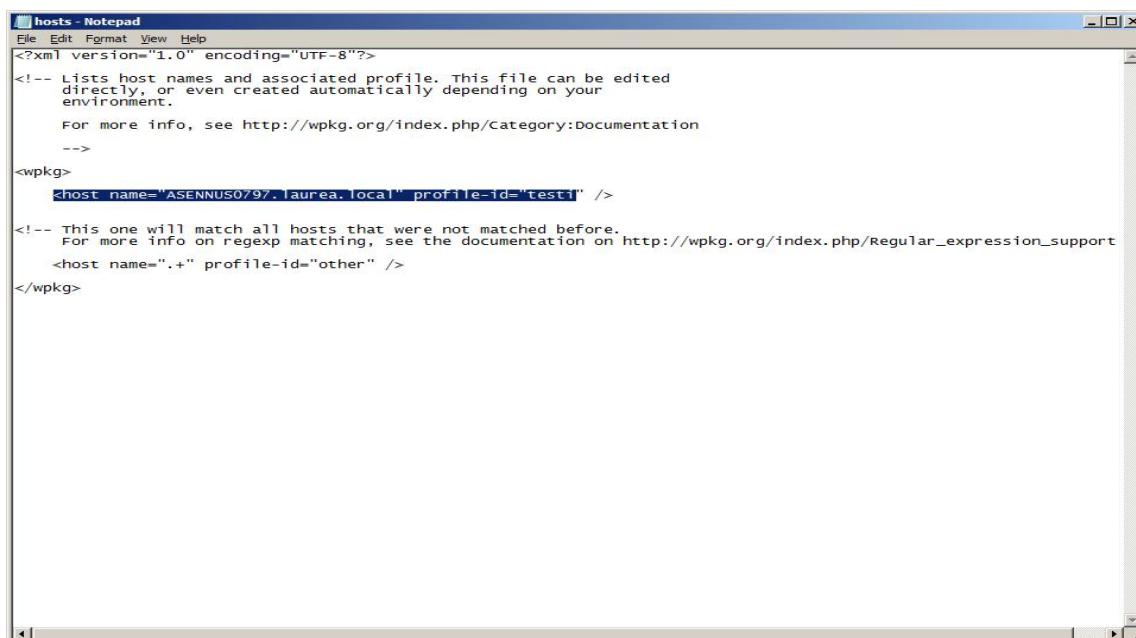
Kuva 4. WPKG:n pääkansio.



Kuva 5. WPKG:n palvelinkansio.

WPKG:n pääkansioista kopioidaan kuvassa 5 näkyvät hosts.xml, packages.xml ja profiles.xml sekä wpkg.js-tiedostot omaan kansioonsa. Nämä kolme xml-tiedostoa käytännössä määrittelevät, mitä ohjelmistopaketteja jaetaan ja mihinkin. Seuraavissa kuvissa kuvataan tiedostoja.

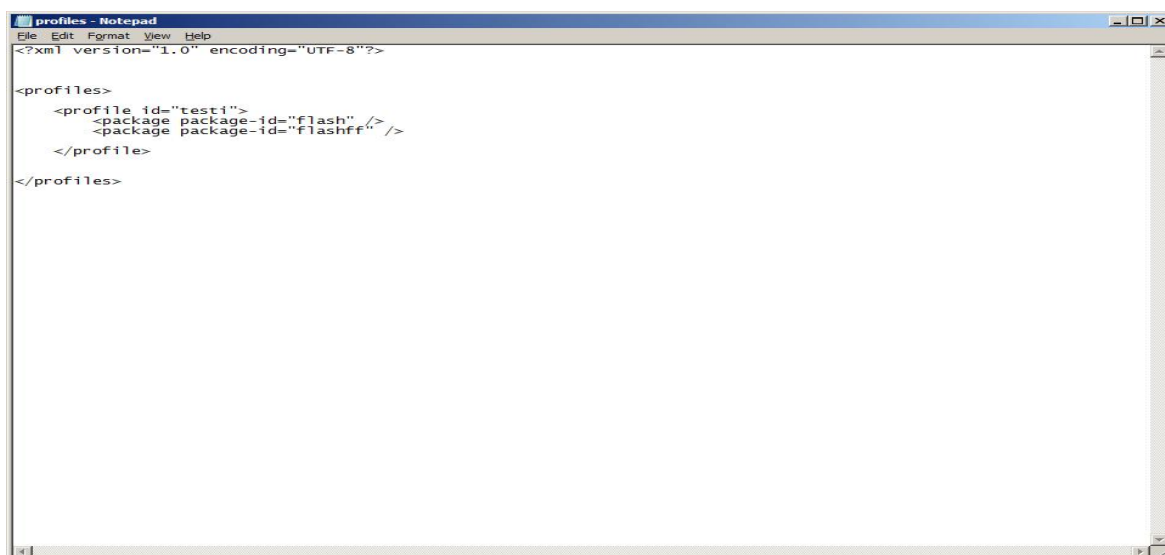
Kuvassa 6 olevassa hosts-tiedostossa määritetään palvelimen host name eli isäntänimi. Isäntänimeksi on määritetty tietokoneen nimi, joka toimii palvelimena. Isäntänimeen lisäksi kytetään profiili, jota tarvitaan profiles-tiedostossa.



```
hosts - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<!-- Lists host names and associated profile. This file can be edited
directly, or even created automatically depending on your
environment.
For more info, see http://wpkg.org/index.php/Category:documentation
-->
<wpkg>
  <host name="ASENNUS0797.laurea.local" profile-id="testi" />
<!-- This one will match all hosts that were not matched before.
For more info on regexp matching, see the documentation on http://wpkg.org/index.php/Regular_expression_support
<host name="." profile-id="other" />
</wpkg>
```

Kuva 6. Hosts.xml.

Kuvan 7 profiles-tiedostossa yksinkertaisesti määritellään, mitkä paketit annetaan millekin profiilille. Tässä tapauksessa testi-profiili saa kaksi pakettia: "flash" ja "flashff". Tässä on hyvä huomata, että hosts-tiedostossa on määritelty sama testi-profiili, jolloin palvelimena toimivalta koneelta saadaan profiilitiedoston kautta määriteltyä asennettavat paketit.



```
profiles - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<profiles>
  <profile id="testi">
    <package package-id="flash" />
    <package package-id="flashff" />
  </profile>
</profiles>
```

Kuva 7. Profiles.xml.

```

packages - Notepad
File Edit Format View Help
<packages>
  <package
    id="Flash"
    name="Adobe Flash Player IE:lle"
    revision="%version%"
    reboot="false"
    priority="10">
    <variable name="version" value="10.2.152.26" />
    <check type="uninstall" condition="exists" path="Adobe Flash Player 10 Activex" />
    <check type="logical" condition="or">
      <check type="file" condition="versionequalto" path="%SYSTEMROOT%\system32\Macromed\Flash\F1ash10m.ocx" value="%version%" />
      <check type="file" condition="versionequalto" path="%SYSTEMROOT%\Syswow64\Macromed\Flash\F1ash10m.ocx" value="%version%" />
    </check>
    <install cmd="%SOFTWARE%\Flash\uninstall_flash_player.exe -uninstall"><exit code='any' /></install>
    <!-- use "/s" or "/silent" as silent-switch if using an older uninstaller-->
    <install cmd="taskkill /f /im iexplore.exe"><exit code='0' /><exit code='1' /><exit code='128' /></install>
    <install cmd="%SOFTWARE%\Flash\install_flash_player_ax.exe /install" />
    <upgrade cmd="%SOFTWARE%\Flash\uninstall_flash_player.exe -uninstall"><exit code='any' /></upgrade>
    <!-- use "/s" or "/silent" as silent-switch if using an older uninstaller-->
    <upgrade cmd="taskkill /f /im iexplore.exe"><exit code='0' /><exit code='1' /><exit code='128' /></upgrade>
    <upgrade cmd="%SOFTWARE%\Flash\install_flash_player_ax.exe /install" />
    <remove cmd="%COMSPEC% /C if exist "%SYSTEMROOT%\system32\Macromed\Flash\F1ashutil10m_Activex.exe" "%SYSTEMROOT%\system32\Macromed\Flash\F1ashutil10m_Activex.exe" /uninstall />
    <remove cmd="%COMSPEC% /C if exist "%SYSTEMROOT%\Syswow64\Macromed\Flash\F1ashutil10m_Activex.exe" "%SYSTEMROOT%\Syswow64\Macromed\Flash\F1ashutil10m_Activex.exe" /uninstall />
  </package>
  <package
    id="Flashff"
    name="Adobe Flash Player Firefox/muut selaimet"
    revision="%version%"
    reboot="false"
    priority="10">
    <variable name="version" value="10.2.152.26" />
    <check type="uninstall" condition="exists" path="Adobe Flash Player 10 Plugin" />
    <check type="logical" condition="or">
      <check type="file" condition="versionequalto" path="%SYSTEMROOT%\system32\Macromed\Flash\NPSWF32.dll" value="%version%" />
      <check type="file" condition="versionequalto" path="%SYSTEMROOT%\Syswow64\Macromed\Flash\NPSWF32.dll" value="%version%" />
    </check>
    <install cmd="taskkill /f /im firefox.exe"><exit code='0' /><exit code='1' /><exit code='128' /></install>
    <install cmd="%SOFTWARE%\Flash\install_flash_player.exe /install" />
    <upgrade cmd="taskkill /f /im firefox.exe"><exit code='0' /><exit code='1' /><exit code='128' /></upgrade>
    <upgrade cmd="%SOFTWARE%\Flash\install_flash_player.exe /install" />
    <remove cmd="%COMSPEC% /C if exist "%SYSTEMROOT%\system32\Macromed\Flash\F1ashutil10m_Plugin.exe" "%SYSTEMROOT%\system32\Macromed\Flash\F1ashutil10m_Plugin.exe" /uninstall />
    <remove cmd="%COMSPEC% /C if exist "%SYSTEMROOT%\Syswow64\Macromed\Flash\F1ashutil10m_Plugin.exe" "%SYSTEMROOT%\Syswow64\Macromed\Flash\F1ashutil10m_Plugin.exe" /uninstall />
  </package>
</packages>

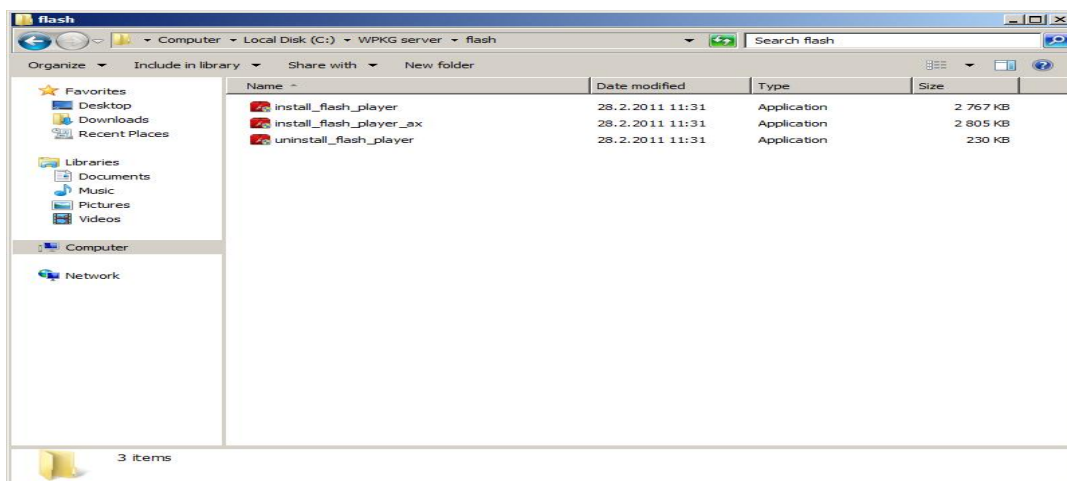
```

Kuva 8. Packages.xml.

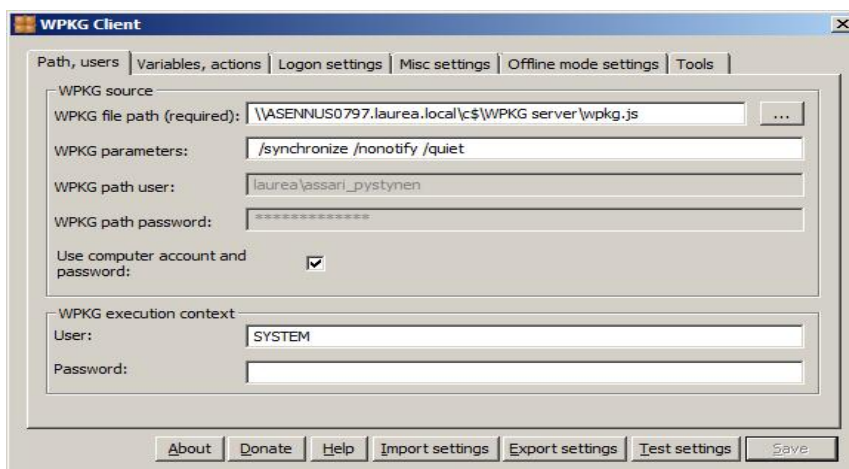
Kuvan 8 packages-tiedosto on WPKG:n xml-tiedostoista monimutkaisin. Tiedostoon voidaan määrittellä useita eri paketteja, joiden id-arvojen tulee olla samat kuin profiles-tiedostoon määritellyt pakettien id-arvot.

Kuvassa 8 on tiedostoon määritelty kaksi eri pakettia. Ensimmäinen paketti asentaa, poistaa tai päivittää Adoben Flash Playerin. Paketin skripti näyttää melko monimutkaiselta, mutta onneksi WPKG:n sivuilta (WPKG.org) löytyy hyvät ohjeet, joiden avulla skriptien lukeminen helpottuu merkittävästi. Lyhyesti selvitettynä kyseinen skripti tarkistaa, löytyykö työasemalta valmiiksi vanhempi versio Flashista. Mikäli vanhempi versio löytyy, poistetaan se ennen uuden version asennusta. Skriptissä tarkistetaan myös onko käyttäjärjestelmä 64- vai 32-bittinen. Tarvittaessa Internet Explorer myös sulkeutuu itsestään ja siihen päivitetään Activex-liitännäiset. Toinen skripti tekee käytännössä saman asian, mutta se on tarkoitettu Mozillan Firefoxille ja muille selaimille.

Paketteihin on myös määritelty, mistä löytyvät tiedostot, jotka asennetaan käyttäjien työasemille. Kuvassa 9 on luotu kansio, johon on ladattu tarvittavat Adoben Flash Playerin asennus sekä poisto-operaatioon tarvitsemat asennustiedostot.



Kuva 9. Adobe Flash Playerin asennustiedostot.



Kuva 10. WPKG Client.

Kuvassa 10 näkyy WPKG:n Client eli asiakasohjelma. Asiakasohjelma asennetaan kaikille koneille, joille halutaan asentaa sovelluksia WPKG:n palvelimelta. Palvelimen osoite tulee kirjoittaa WPKG file path -kohtaan. Asiakasohjelmalle kerrotaan tässä, mistä se löytää wpkg.js-tiedoston, joka osaa lukea kansiossaan olevat xml-määrittelytiedostot. Xml-tiedostot on kuvattu kuvissa 6-8.

Asiakasohjelmalle annetaan lisäksi tarvittavat asennusparametrit, tässä tapauksessa on tyydytty ohjelman valmiiksi antamiin oletusasetuksiin. WPKG path user ja password-kohtiin tulee

kirjoittaa käyttäjätunnus, jolla on riittävät oikeudet sovelluksen asentamiseen koneelle. Muista välilehdistä saadaan säädettyä asennukseen liittyviä asetuksia.

Tärkeitä painikkeita kuvan 10 alareunassa ovat Export settings ja Save. Export settings -painikkeella saadaan luotua asetustiedosto palvelinkoneella, josta se saadaan tuotua (engl. import) asiakaskoneille asiakasohjelman kautta tai vaihtoehtoisesti komentoriviltä asiakasohjelman asennuksen yhteydessä.

7.4.2 Ajatuksia WPKG:n käytöstä Laureassa

WPKG:n käyttöönotto on kohtalaisen yksinkertaista WPKG:n internetsivustolta löytyvien hyvien ohjeiden ansiosta. Ohjelmalla saadaan tehtyä valmiita asennuspaketteja helposti yhdistelemällä useita eri asennuskriptejä. Ohjelman käyttöönottoa helpottaa merkittävästi se, että ohjelman sivustolta löytyy kaikkien yleisimpien ohjelmien asennuskriptit valmiina. Pienellä muokkauksella ja lisäämällä useampia skriptejä yhteen pakettiin saadaan luotua suhteellisen pienellä vaivalla valmiita asennuspaketteja. Valmiista skripteistä löytyy myös esimerkiksi käyttöjärjestelmän bittisyyden tarkistukset ja muut hankalahkot paketin toimintavarmuutta ja yhteensopivuutta parantavat asiat.

WPKG:lla luodut sovelluspaketit on helppo jakaa muutamille koneille. Laajamittaisten sovellusjakeluiden suorittamiseen käyvät helpommin MSI-paketit, jotka voidaan jakaa suoraan aktiivihakemiston kautta. WPKG:lla tehtyjä paketteja pystytään jakamaan myös aktiivihakemiston kautta, mutta sivustolta löytyvät ohjeet olivat testien aikana saksaksi ja vaikuttivat muutenkin jokseenkin sekavilta, joten aktiivihakemisto päätettiin jättää pois tämän ohjelman testauksesta kokonaan. Sen sijaan pakettien jakelu toimii helposti pienelle määrälle koneita määrittelemällä asiakaskoneille palvelinkoneen, josta paketit asennetaan. Tämän jälkeen asiakaskoneiden määrää on nopea lisätä tarvittaessa. Kuitenkin laajempi jakelu kannattaa suorittaa toisella tapaa.

WPKG:lla tehdyt paketit toimivat huomattavasti varmemmin kuin EMCON MSI-paketit. Ohjelmaa voitaisiinkin käyttää Laureassa esimerkiksi joidenkin projektikoneiden tai muun pienen koneryppään sovelluspaketointiin ja -jakeluun. Paneutumalla ohjelmaan tarkemmin siitä saa melko varmasti myös työkalun laajamittaisempaankin sovellusjakeluun.

Suurimmat ongelmatilanteet WPKG:n käytössä ovat käytännössä samanlaisia kuin tavallisia asennuskriptejä tehdessäkin. Väärät hakemistopolut tai kirjoitusvirheet aiheuttavat suurimman osan ongelmista. Lisäksi varsinkin ohjelmaan tutustuessa ilmenee jonkin verran ongelmia, sillä pitää seurata samaan aikaan useaa eri xml-tiedostoa, jotka sisältävät runsaasti tietoa.

7.5 Asennuspaketin luominen EMCO MSI Package Builderilla

MSI-paketointia päädyttiin kokeilemaan EMCO MSI Package Builderilla. Ohjelmisto on helppokäyttöinen ja suurin osa ohjelman toiminnoista onnistuu varsin luontevasti heti ensimmäisellä käyttökerralla. Ohjelman ehdoton miinuspuoli onkin sen kaupallisuus. Ohjelman yrityslicenssi täydellisestä Enterprise-versiosta maksaa noin 450 euroa. Tässä projektissa päädyttiin käyttämään ohjelman kokeiluversiota. Kokeiluversiokin tosin mahdollistaa ohjelman ajamisen kolmekymmentä kertaa, joten riippuen paketointikoneen uudelleenkäynnistystahdistista ohjelmaa voidaan käyttää melko pitkään asennuspakettien luomiseen. Tätäkin rajoitusta voidaan tosin kiertää helposti asentamalla ohjelma toiselle koneelle tai uudestaan mikäli rajoitukset tulevat vastaan. Tähän projektiin kyseinen rajoitus ei kuitenkaan vaikuta negatiivisesti.

7.5.1 EMCO MSI Package Builderin käyttöönotto

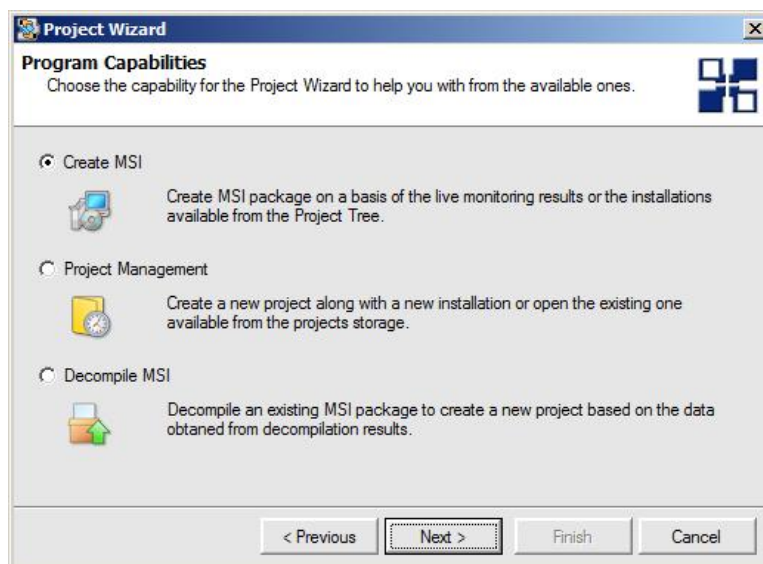
Ohjelman käyttöönotto on varsin yksinkertainen operaatio. Aluksi ohjelma ladataan ja asennetaan halutulle työasemalle, jolla paketointi suoritetaan. Kun asennus on valmis, voidaan ohjelma käynnistää. Ohjelman käynnistyessä ruudulle ponnahtaa kuvassa 11 näkyvä ruutu.



Kuva 11. MSI Package Builderin aloitusruutu.

Tässä tapauksessa valitaan kohta Evaluate, sillä tässä vaiheessa ei haluta maksaa lisenssimaksuja. Mikäli lisenssiä tarvitaan, voidaan se syöttää kuvassa 11 näkyvässä ruudussa Enter License kohdassa.

Seuraavassa kohdassa valitaan ohjelman käyttämä projektien oletuskansio, johon tallennetaan ohjelmassa luodut projektit. Kun kansio on määritelty, ohjelma tarjoaa projektivelhon (engl. Project Wizard). Kuvassa 12 näkyvät samalla ohjelman päätoiminnot.



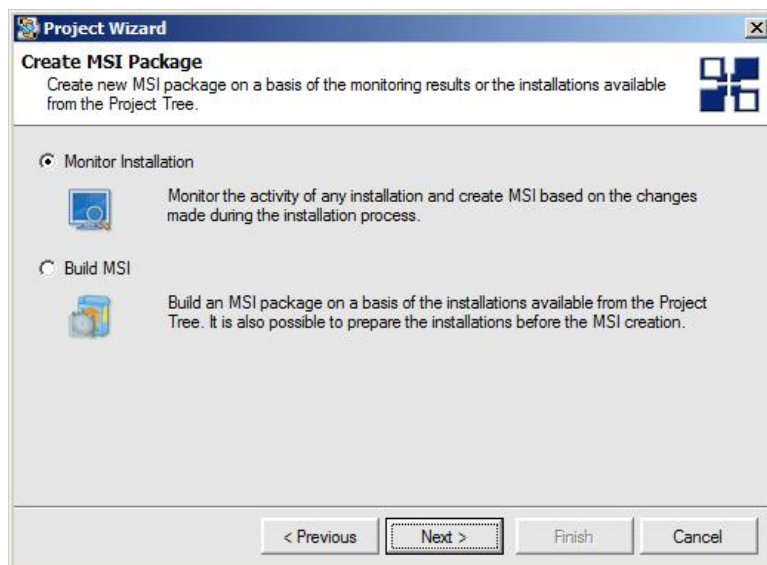
Kuva 12. MSI Package Builderin Projektivelho.

Create MSI -kohdassa ohjelmalla voidaan luoda MSI-paketti, joka on luotu projektissa käyttäen nauhoitusmenetelmää (engl. Monitoring), eli ohjelma on nauhoittanut kaikki tietokoneella tapahtuneet muutokset halutulla aikavälillä. Käytännössä muutoksiin kuuluvat ohjelmien poistot, muokkaukset ja asennukset. Tarvittavat rekisterimerkinnot luodaan samalla automaattisesti.

Project Management tarkoittaa yksinkertaisesti projektinhallintaa. Voidaan luoda kokonaan uusi projekti tai avata vanha projekti, johon mahdollisesti voidaan luoda uusia paketteja.

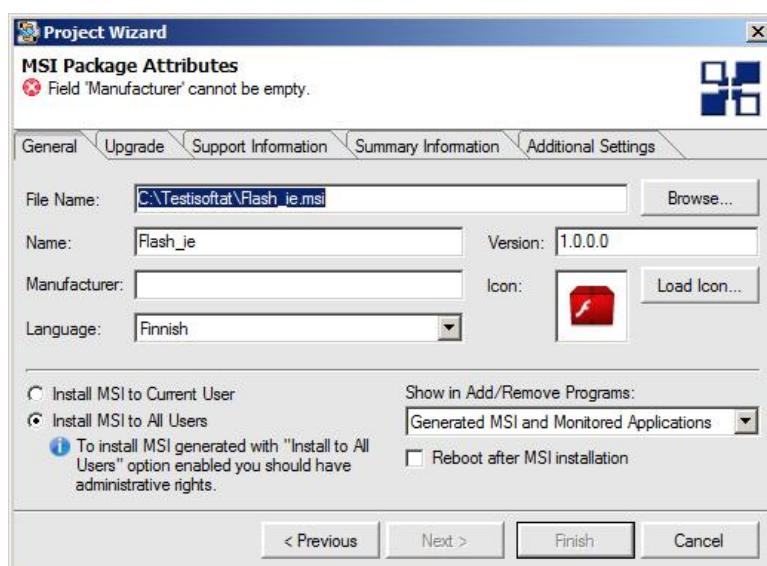
Decompile MSI -toiminnolla luodaan projekti, joka pohjautuu jo olemassa olevaan MSI-paketissa olevaan tietoon. Eli toiminnolla valitaan MSI-paketti, jonka pohjalta voidaan luoda uusi projekti.

Valitaan kohta Create MSI, jolloin ruutuun ilmestyy kuvan 13 kaltainen näkymä. Vaihtoehdot ovat Monitor Installation sekä Build MSI. Jälkimmäinen vaihtoehto valitaan, mikäli halutaan luoda MSI-paketti, joka pohjautuu johonkin aiemmin luotuun projektiin. Tässä tapauksessa kuitenkin halutaan luoda uusi MSI-paketti, jolloin valitaan Monitor Installation.



Kuva 13. Projektivelhon MSI-paketin luonti.

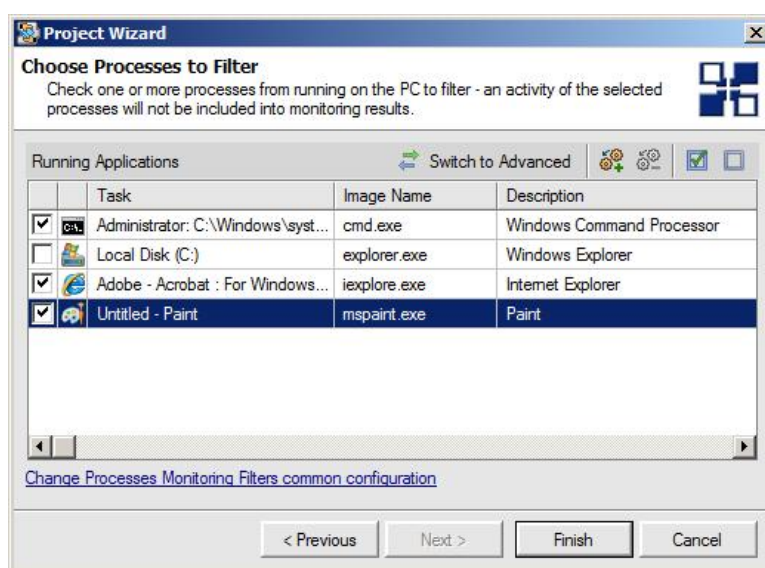
Seuraavaksi Projektivelho pyytää asennustiedoston sijainnin, josta MSI-paketti luodaan ja projektin nimen. Painamalla Next päästään kuvan 14 näkymään. File Name -kohtaan ilmestyy automaattisesti tulevan MSI-paketin sijainti ja nimi. Tästä ruudusta on tärkeää valita Install MSI to All Users, jolloin MSI-paketti asentuu kaikille työaseman käyttäjille. Myös Show in Add/Remove Programs -kohdasta kannattaa vaihtaa Monitored Applications only, jolloin Windowsin ohjelmien lisäys- ja poistovalikkoon ei ilmesty valmistuvaa MSI-pakettia, vaan pelkästään työasemalle asentuvat ohjelmat.



Kuva 14. Projektivelhon MSI-paketin ominaisuudet.

Additional Settings -välilehdeltä voidaan vaihtaa tuetut käyttöjärjestelmät ja asennusehdot, kuten asentuuko ohjelma jos työasemalta löytyy jo valmiiksi saman ohjelman sama tai uudempi versio. On mahdollista myös pakottaa ohjelma asentumaan, jolloin ohjelma asentuu versiosta riippumatta. Tämä saattaa kuitenkin aiheuttaa ongelmia ohjelman toiminnassa, mikäli yritetään asentaa vanhempaa versiota uuden päälle poistamatta ensiksi edellistä.

Kohdasta Next päästään valitsemaan, mitkä prosessit jätetään Monitoring-menetelmän ulkopuolelle. Tällä siis tarkoitetaan sitä, että valittujen sovellusten toimet eivät vaikuta MSI-paketin sisältöön millään tavalla. Advanced kohdasta voidaan myös yksitellen määrittellä esimerkiksi rekisterimerkintöjen suodatusta.

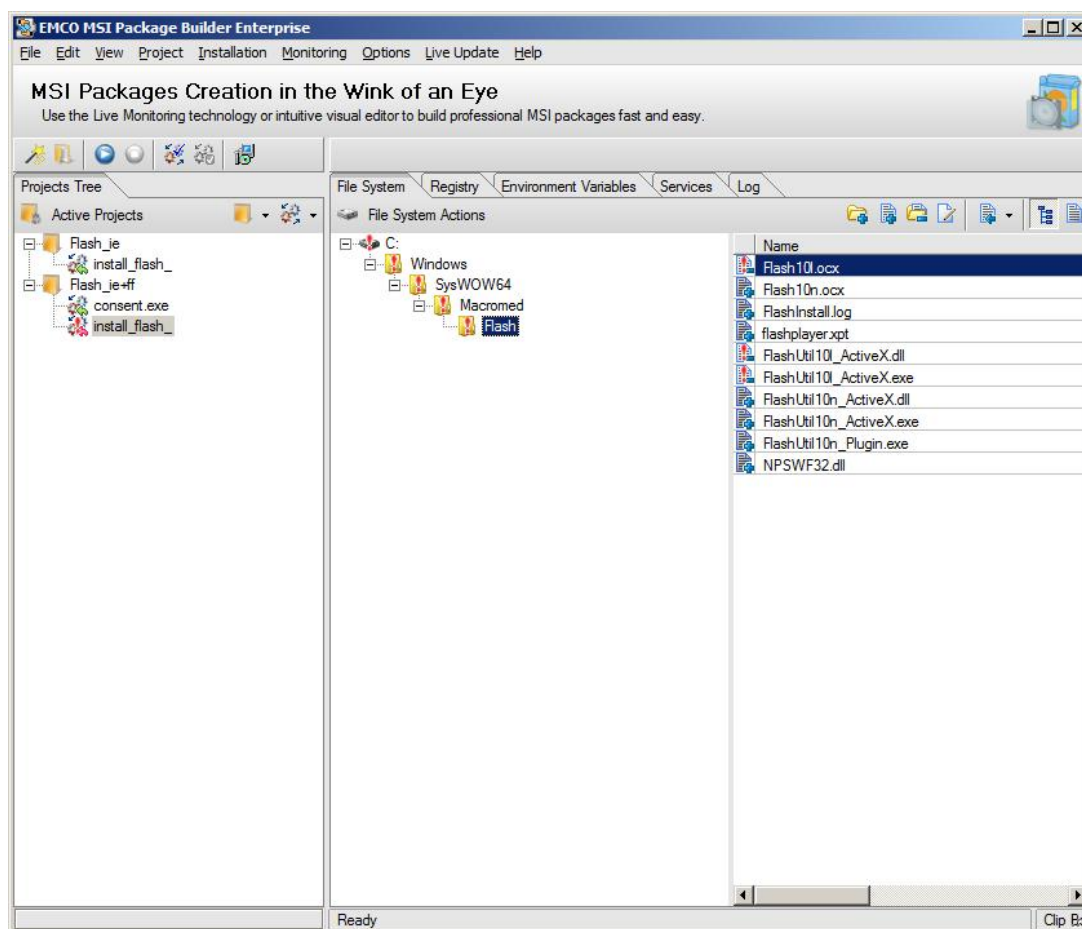


Kuva 15. Ylimääräisten prosessien suodatus.

Painamalla Finish luodaan asetusten mukainen MSI-paketti kuvassa 14 määriteltyyn sijaintiin. Kun paketti on luotu, avautuu ohjelman pääikkuna, joka näkyy kuvassa 16. Ruudun vasemmassa laidassa, kohdassa Projects Tree näkyvät kaikki luodut projektit, joiden alla sijaitsevat monitoring-menetelmällä nauhoitetut asennukset.

Kuvan keskeltä voidaan tarkastella, mitä muutoksia työasemalle tapahtui nauhoituksen aikana. File System -kohdasta ilmenevät tiedostojärjestelmään tulleet muutokset sekä Registry-kohdasta näkyvät rekisteriin tapahtuneet muutokset. Tarpeen mukaan näihin merkintöihin voidaan tehdä muutoksia ennen kuin lopullinen MSI-paketti luodaan.

Environment Variables -kohtaan voidaan määrittellä erilaisia muuttujia, Services välilehdellä puolestaan pystytään manuaalisesti määrittelemään eri palveluita käynnistymään. Log-kohdassa puolestaan ilmoitetaan asennuksen aikana tapahtuneista virheilmoituksista ja esimerkiksi suodatetuista prosesseista, jotka jätettiin nauhoituksen ulkopuolelle.



Kuva 16. Ohjelman pääikkuna.

Projektivelholla on helppo luoda MSI-paketti, mutta projektivelhon käyttäminen sekavoittaa ohjelman käyttölogiikkaa. Ohjelma antaa eri järjestyksessä eri valintaruudut riippuen tavasta, jolla paketti tehdään. Ohjelma on helppokäyttöinen, kun sovelluspaketin luomisprosessi aloitetaan painamalla Start Monitoring -painiketta. Ohjelma kysyy vain projektin nimen, jonka jälkeen voidaan valita nauhoituksessa suodatettavat prosessit. Tämän jälkeen päästään takaisin ohjelman pääikkunaan. Nauhoitus on käynnissä niin pitkään kunnes ohjelmaa käsketään lopettamaan nauhoitus painamalla Stop Monitoring -painiketta. Nauhoituksen aikana täytyy muistaa, että ei tehdä mitään ylimääräistä, sillä kaikki toiminnot tallentuvat, lukuun ottamatta niitä sovelluksia, jotka on suodatettu pois. Kun nauhoitus on pysäytetty, ilmestyy ohjelman pääikkunaan aktiivisen projektin alle kaikki muutokset, jotka tapahtuivat nauhoituksen aikana.

Aktiivisesta projektista voidaan luoda MSI-paketti painamalla Create MSI Package -painiketta. Tällöin avautuu vastaava ruutu kuin kuvassa 14. Toisin kuin projektivelhossa, täytyy paketille itse määrittellä sijainti eli mihin paketti luodaan. Muuten toimitaan vastaavasti kuin projektivelhossa.

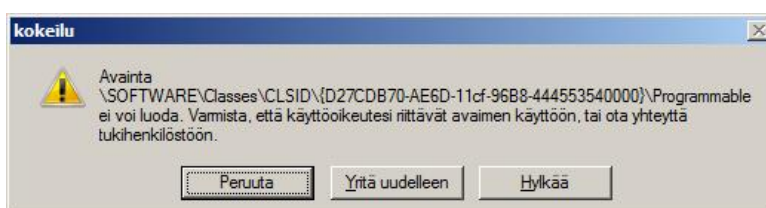
Kun asennuspaketti on valmis, voidaan se jakaa käyttäjien koneille käyttäen Windowsin aktiivihakemiston ryhmäkäytänteitä.

7.5.2 Ajatuksia EMCO MSI Builderin käytöstä Laureassa

EMCO MSI Builderin käyttöönotto on vaivatonta. Lisäksi ohjelmalla saa tehtyä melko suurella todennäköisyydellä toimivan MSI -paketin. Käytännössä ohjelmalla tehdyt paketit toimivat kuitenkin vain harvoin täydellisesti. Koska MSI-pakettiin nauhoitetaan kaikki nauhoituksen aikana tehdyt muutokset, saattaa pakettiin jäädä kaikista varotoimista huolimatta ylimääräistä roskaa, joka haittaa koko asennuspaketin toimivuutta. Lisäksi eri ohjelmistoversiot aiheuttavat runsaasti päänvaivaa. Paketit toimisivat varmasti huomattavasti paremmin, mikäli kaikissa työasemissa olisi täsmälleen samat versiot kaikista ohjelmista. Tämän lisäksi tulisi ottaa huomioon eri käyttöjärjestelmät. Vanhat Windows XP:lle tehdyt paketit eivät toimi suoraan esimerkiksi Windows 7:ssä. Vielä kun otetaan käyttöjärjestelmän bittisyys huomioon, alkaa todennäköisyys toimivalle paketille huonontua entisestään. Käytännössähän tämä tarkoittaa sitä, että kaikille käyttöjärjestelmäversioille tulee tehdä omat sovelluspaketit, joka taas aiheuttaa runsaasti lisävaivaa ja ohjelman käyttöönotossa säästetty aika saattaa kostautua pian, kun ohjelmaa joutuukin käyttämään kymmenien pakettien luomiseen. Eräs erittäin tärkeä asia on myös pakettien päivitettävyyden. Kun MSI-paketteja muokataan, tulee taas ottaa huomioon edellä mainitut ongelmakohdat.

Ohjelmalla luotuja MSI-paketteja asennettaessa niitä on ilmestynyt mitä kummallisimpia virheilmoituksia. Luotaessa paketteja yleisimmät virheet johtuvat yleensä joko ylimääräisten sovellusten vaikutuksesta tai käyttöoikeuksien puuttumisesta. Käyttöoikeuksien puute johtuu Windows 7 -ympäristössä hyvin isolla todennäköisyydellä Windowsin käyttäjätilien valvonnasta (engl. User Account Control), joka pyytää lupaa sovellusten asentumiseksi.

Johtuen mahdollisesti juuri käyttäjätilien hallinnasta aiheutuneesta virheestä valmiiseen pakettiin ei työasemalle ole oikeutta asentaa sovelluspakettia. Kuvassa 17 näkyvä virheilmoitus on yleinen varsinkin yrittäessä asentaa Adoben Flash Playerin MSI-pakettia Windows 7 -koneelle.



Kuva 17. Esimerkki virheilmoituksesta.

Ohjelmalla luodut MSI-paketit on helppo jakaa aktiivihakemiston kautta, mutta tämän helpouden takeena ovat toimivat paketit. Rikkinäisiä paketteja on turha jaella mihinkään. Teoriassa kuitenkin tällä ohjelmalla pystyisi hoitamaan koko Laurean sovelluspaketoinnin sekä niiden jakelun aktiivihakemiston kautta. Valitettavasti kuitenkin pakettien toimivuuden epävarmuuden takia ei ainakaan näiden testien tekijä pysty suosittelemaan ohjelman käyttöä, ainakaan laajassa mittakaavassa.

Ohjelmaa voitaisiin ehkä käyttää pakettien luomiseen ja sitten muokata niitä jollakin toisella ohjelmalla. Tästä herääkin kysymys, onko järkevää maksaa tähän käyttötarkoitukseen suunnitellusta ohjelmasta, kun ilmaisiakin vaihtoehtoja on saatavilla. Mikäli MSI-paketteja halutaan luoda Laurean työympäristössä, saattaakin olla järkevää opetella hieman monimutkaisemman ohjelman käyttö, jolla pystytään tekemään mitä halutaan.

8 Johtopäätökset

Tämän opinnäytetyön tavoitteina oli tutkia eri paketointimenetelmiä ja löytää niistä paras, jota voitaisiin käyttää Laureassa helpottamaan ylläpitäjien työrutiineja. Eri paketointimenetelmillä on omat heikkoutensa ja vahvuutensa, joten suoraa vastausta kysymykseen parhaasta paketointimenetelmästä on vaikea sanoa. Projektiosiossa käytössä olleella EMCO ohjelmistolla onnistui sovelluspakettien luonti vaivattomasti. Ohjelmalla luotujen MSI-pakettien jakelu onnistuu helposti, mutta tämän projektin tulosten perusteella pakettien luomisprosessi on liian epävarma, jotta ohjelman käyttöönottoa voitaisiin vakavasti harkita.

Toinen projektissa testauksessa ollut ohjelmisto WPKG antoi huomattavasti positiivisemmän kuvan itsestään. Pakettien luominen onnistui huomattavasti varmemmin kuin EMCO:lla luodut MSI-paketit. WPKG:n ainoa rajoittava tekijä on aktiivihakemiston kautta suoritettavan ohjelmistojakelun suorittaminen, joka vaatii huomattavan määrän perehtymistä aiheeseen.

Nykyisten Laureassa käytössä olevien paketointimenetelmien tukena WPKG voisi toimia erinomaisesti. Varsinkin hieman valtavirrasta poikkeavien koneryhmien sovellusjakeluun se vaikuttaisi toimivan hyvin. WPKG myös tarjoaa valmiita skriptejä, joista on hyötyä tehdessä omia asennusskriptejä. WPKG voisi olla tulevaisuudessa varteenotettava vaihtoehto varsinkin silloin, kun on selvitetty, kuinka aktiivihakemiston kautta WPKG:n paketteja pystytään jakamaan.

Tulevaisuudessa Laureassa voitaisiin kokeilla toisia paketointimenetelmiä. Esimerkiksi snapshot-periaattella toimivat MSI-paketointiohjelmat voisivat olla kokeilemisen arvoisia juu-

rikin MSI-pakettien jakelun helppouden takia. Tässäkin on tosin samat riskit kuin monitoring-menetelmässä, joten järkevintä lienee pysytellä asennusskriptien puolella, sillä ne toimivat täsmälleen kuten niiden käsketään toimivan. Inhimillisiä virheitä niitakin tehdessä toki sattuu, mutta sen takia skriptit tulee tarkistaa kahteen kertaan. Vaihtoehtoisesti voidaan luoda omia MSI-paketteja tai muokata valmistajien valmiita paketteja, jolloin niitä voidaan räätälöidä hyvin omiin tarpeisiin.

Loppuyhteenvedon voidaan todeta, että MSI-pakettien luominen maksullisella ohjelmalla on helppoa, mutta paketit toimivat epävarmasti. MSI-pakettien luominen ja muokkaus ilmaisilla työkaluilla on taas työlästä ja vaatii asiantuntijuutta. Asennusskriptit toimivat varmasti, mutta monimutkaisten skriptien luominen on työlästä. Parhain ratkaisu onkin miettiä keinoja, joilla pystytään yhdistelemään näitä menetelmiä. Tämän pohdinnan perusteella ehdottomasti WPKG on tutkimisen arvoinen ohjelma, josta pienellä lisätutkimuksella voi olla runsaasti iloa Laurean ylläpitäjille tulevaisuudessa.

Lähteet

AppDeploy 2011. AppDeploy Repackager. Viitattu 2.3.2011.
<http://www.appdeploy.com/tools/repackager/>

Bareis 2011. MAKEMSI. Viitattu 4.3.2011.
<http://dennisbareis.com/makemsi.htm>

Brothersoft 2008. MSIBuilder. Viitattu 1.3.2011.
<http://www.brothersoft.com/msibuilder-20541.html>

Emco Software 2011. EMCO MSI Builder. Viitattu 15.3.2011.
<http://emcosoftware.com/products/msi-package-builder/features.php>

Juntunen, A. 2007. Sovellusten MSI - paketointi ja jakelu. Viitattu 12.3.2011.
<http://publications.theseus.fi/bitstream/handle/10024/7043/KAT4TAKiJ.pdf?sequence=1>

Microsoft Corporation 2009. Enabling Server and Desktop Virtualization. Viitattu 12.2.2011.
<http://www.microsoft.com/windowsserver2008/en/us/r2-virtualization.aspx>

Microsoft Corporation 2010. MSDN Library, Windows Installer. Viitattu 19.3.2011.
[http://msdn.microsoft.com/en-us/library/cc185688\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc185688(VS.85).aspx)

Microsoftin tuotetuki 2010. MSIOrca. Viitattu 3.3.2011.
<http://support.microsoft.com/kb/255905>

Ruuska, J. 2006. Windows Installer - asennuspakettien luonti, muokkaus ja jakelu. Viitattu 20.3.2011.
<https://publications.theseus.fi/bitstream/handle/10024/10319/TMP.objres.476.pdf?sequence=2>

Tulloch 2005. MSI Packaging Tools. Viitattu 20.12.2010.
http://www.windownetworking.com/articles_tutorials/MSI-Packaging-Tools.html

Teknologian tutkimuskeskus VTT. Riskianalyysit. Viitattu 8.5.2011
<http://virtual.vtt.fi/virtual/riskianalyysit/index53d4.html>

WPKG. WPKG paketoitsovellus. Viitattu 21.2.2011.
<http://wpkg.org/>

Kuvat ja kuvat

Kuva 1. Hyper-V Managerin pääikkuna.	18
Kuva 2. Uuden virtuaalikoneen määritykset valmiina.	19
Kuva 3. Windowsin asennus.	20
Kuva 4. WPKG:n pääkansio.	24
Kuva 5. WPKG:n palvelinkansio.	24
Kuva 6. Hosts.xml.	25
Kuva 7. Profiles.xml.	25
Kuva 8. Packages.xml.	26
Kuva 9. Adobe Flash Playerin asennustiedostot.	27
Kuva 10. WPKG Client.	27
Kuva 11. MSI Package Builderin aloitusruutu.	29
Kuva 12. MSI Package Builderin Projektivelho.	30
Kuva 13. Projektivelhon MSI-paketin luonti.	31
Kuva 14. Projektivelhon MSI-paketin ominaisuudet.	31
Kuva 15. Ylimääräisten prosessien suodatus.	32
Kuva 16. Ohjelman pääikkuna.	33
Kuva 17. Esimerkki virheilmoituksesta.	34

Taulukot

Taulukko 1. Riskianalyysin osa-alueet pisteytettynä.	22
---	----