

L U K U 14

Sovellusten jakelu

Oppitunti 1: Windows asennusohjelmat 576

Oppitunti 2: InstallShield 585

Oppitunti 3: COM-komponenttien rekisteröinti asiakaskoneessa 591

Oppitunti 4: Jakeluvaihtoehdot 596

Oppitunti 5: Zero Administration for Windows 603

Laboratorio 14: STUload-sovelluksen paketointi ja jakelu 607

Kertaus 612

Tässä luvussa

Kun sovellus on testattu, jäljellä on vielä yksi tehtävä: jakelu käyttäjille. Tehtävä ei ole niin helppo kuin voisi kuvitella, jakelu vaatii pohdintaa ja suunnittelua.

Tässä luvussa kerrotaan muutamia tapoja, joilla Microsoft Visual C++:lla tehtyjä Windows-sovelluksia voidaan levittää tehokkaasti. Tässä luvussa tehdään ensin yleiskatsaus jakelumenetelmiin. Opit, kuinka käytetään InstallShieldiä, työkalua, joka lähes täysin automatisoi asennusohjelman tekemisen Visual C++-projekteille. Opit myös, kuinka asennusohjelma asentaa ActiveX-kontrollit ja muu COM-komponentit, ja kuinka Microsoft Zero Administration for Windows (ZAW) tulee vaikuttamaan ohjelmien asentamiseen tulevaisuudessa.

Ennen kuin aloitat

Ennen kuin aloitat tämän luvun läpikäymisen, sinun tulisi lukea luvut 2-13 ja tehdä niihin kuuluvat tehtävät.

Oppitunti 1: Windows-asennusohjelmat

Jakelu koostuu kahdesta vaiheesta. Ensin täytyy siirtää tarvittavat tiedostot käyttäjän kiintolevylle. Tämän jälkeen seuraa toinen, ehkä vaikeampi vaihe: isäntäkone on saatava konfiguroitua niin, että se tunnistaa ja pystyy suorittamaan asennetun sovelluksen. Varmistaaksesi onnistuneen asennuksen sinun täytyy auto-matisoida sovelluksen käyttöönotto mahdollisimman hyvin. Windows-käyttäjät ovat tottuneet siihen, että asennusohjelma tekee suurimman osan työstä heidän puolestaan.

Tämän oppitunnin jälkeen:

- Tiedät, kuinka sovellus asennetaan ja poistetaan käyttäjän järjestelmästä.
- Osaat luetella tavallisen asennusohjelman tarjoamat palvelut.
- Tiedät, mitä merkintöjä asennusohjelma tekee käyttäjän rekisteriin.
- Osaat luoda cabinet ja rekister -tiedostoja.

Oppitunnin kesto: 20 minuuttia

Asennusohjelmissa noudatetut käytännöt

Asennusohjelman nimeksi tulisi aina antaa Setup kahdesta syystä, jotka ovat:

- Yhdenmukainen käytäntö.
- Ohjauspaneelin Lisää/Poista sovellus -toiminnon tunnistuksen helpottaminen.

Yhtenäisestä Setup-tiedoston nimestä on tullut sovelluskehitysohjelmissa yleinen käytäntö, joka on omiaan helpottamaan käyttäjän elämää. Windowsin peruskäyttäjä tietää, kuinka Setup-nimisen ohjelman kanssa toimitaan, vaikka ohjelmat voivatkin erota tyyliltään ja toiminnoiltaan.

Jos sovelluksesi jakelu vaatii useita levykkeitä tai romppuja, asennusohjelman tulisi sijaita paketin ensimmäisellä levykkeellä tai rompulla, jotta Windowsin Lisää/Poista sovellus -toiminnon olisi se mahdollisimman helppo löytää.

Käyttäjälle voidaan tarjota useita asennuspalveluja. Tyypillien asennusohjelma:

- Antaa mahdollisuuden vaikuttaa sovelluksen asennusvaihtoehtoihin.
- Luo vaaditut kansiot käyttäjän kiintolevylle.
- Kopioi tiedostot asennusvälineeltä kiintolevylle.
- Rekisteröi jaetut DLL:t.
- Rekisteröi sovelluksen ActiveX-kontrollit ja muut COM-komponentit.

- Lisää järjestelmärekisteriin tiedon, joka määrittelee sovelluksen poistamiseen tarvittavan komennon. Lisää/Poista sovellus -toiminto hakee sen perusteella sovelluksen asennuksen purkavan ohjelman.
- Lisää sovelluksen dokumenttitiedostotunnisteen rekisteriin niin, että käyttäjä voi avata tiedoston sovelluksessa kaksoisnapauttamalla sen kuvaketta resursienhallinnassa.
- Lisää merkinnän Windowsin **Ohjelmat**-valikkoon tai pikakuvakkeen työpöydälle.
- Ajaa "kerran ajettavan koodin" asennuksen koon moinimomiseksi, jos Setup-ohjelmaa ei kopioida kiintolevylle.
- Lisää tai poistaa valitut toiminnot asennetusta sovelluksesta, kun asennusohjelma ajetaan uudelleen.

Käyttäjän tulisi voida asentaa ohjelma vain valitsemalla asennusvaihtoehto ja vaihtamalla uusia levykkeitä tai romppuja pyydettyä. Microsoft suosittelee, että asennusohjelmassa olisi käytettävissä seuraavat neljä vaihtoehtoa:

- *Compact* vaihtoehto on tarkoitettu kannettaviin tietokoneisiin ja järjestelmiin, joissa levytila on rajoitettu.
- *Custom* antaa käyttäjille mahdollisuuden valita, mitä asennetaan.
- *Typical* on useimmille käyttäjille sopiva vaihtoehto.
- *Silent* on automaattinen asennusvaihtoehto.

Silent asennusta tarvitsevat järjestelmän pääkäyttäjät, jotka haluavat asentaa sovelluksia verkon kautta. Toimiessaan Silent-tilassa asennusohjelman ei tulisi esittää kysymyksiä eikä näyttää virheilmoituksia ja sen täytyy omaksua järkevät oletusasetukset kaikkiin tapauksiin.

Ohjeita asennusohjelman kirjoittamiseen

Pidä seuraavat asiat mielessä suunnitellessasi asennusohjelmaa sovelluksesi jakelua varten. Älykkään asennusohjelman tulisi:

- Varastoida yksityiset .ini-tiedostot sovelluksen hakemistoon, jos sovellus on paikallinen tai **GetWindowsDirectory()** API-funktion palauttamaan hakemistoon, jos sovellus on jaettu.
- Välttää sopimatonta tiedostojen kopioimista Windows, WinNT, System tai System32 -hakemistoihin. Jos sovelluspakettisi sisältää fontitiedostoja, ne tulisi sijoittaa järjestelmän Fonts-kansioon.
- Varmistaa ennen kopioimista, ettei tiedosto jo ole kiintolevyllä. Jos asennusohjelma huomaa päällekkäisyyden, sen tulisi osata päättää, kumpi tiedostoista on uudempi ja välttää uudemman tiedoston korvaamista vanhemmalla.

- Tarjota oletuksia. Erityisesti asennusohjelman tulisi sisältää oletusvastaukset kaikkiin kysymyksiin niin, että käyttäjä voi suorittaa onnistuneen oletusasennuksen vastaamalla kaikkiin kysymyksiin painamalla ENTER-näppäintä.
- Välttää pyytämästä käyttäjältä samaa levyä useampaa kertaa.
- Kertoa käyttäjälle tarvittavan levytilan määrä.
- Näyttää tietoa asennuksen edistymisestä.
- Tallettaa väliaikaiset tiedostot Temp-hakemistoon. Kuitenkin, jos asennusohjelma käynnistää Windowsin uudelleen asennuksen aikana ennen väliaikaisten tiedostojen käsittelyä, tulee varmistua siitä, että Temp-hakemisto sijaitsee kiintolevyllä eikä RAM-levyllä. Näin voidaan olla varmoja, että tiedostot ovat edelleen tallessa uudelleenkäynnistyksen jälkeen.
- Antaa käyttäjälle mahdollisuus perua asennus ennen sen valmistumista. Asennusohjelman tulisi pitää kirjaa kopioiduista tiedostoista ja muutetuista asetuksista, jotta järjestelmä voidaan siivota, mikäli asennus perutaan.

Käyttäjä asentaa sovelluksesi joko ajamalla asennusohjelman suoraan tai käyttämällä Ohjauspaneelin Lisää/poista sovellus -toimintoa. Lisää/poista sovellus -toiminto etsii automaattisesti levykkeeltä tai rompulta Setup.exe-ohjelmaa. Jos tiedosto löytyy ja käyttäjä haluaa suorittaa asennuksen, Lisää/poista sovellus käynnistää Setup-ohjelman ja sulkeutuu. Lisää/poista sovellus -toiminnon sulkeuduttua asennusohjelma on vastuussa käyttäjän opastamisesta asennuksen aikana.

Uninstall-ohjelmien käytännöt

Asennetun sovelluksen tulisi voida turvallisesti ja mahdollisimman täydellisesti poistaa kaikki jäljet itsestään. Tämän toiminnon, joka tunnetaan myös nimityksillä *de-installing* ja *uninstalling*, suorittaa yleensä erillinen ohjelma, joka on nimetty kuvailevalla nimellä kuten Uninstall tai Uninst. Vaihtoehtoisesti voidaan itse Setup-ohjelma ohjelmoida toimimaan asennuksen purkajana. Uninstall-ohjelman nimi on melko vapaasti päätettävissä, koska asennusohjelma tallettaa sen nimen ja sijainnin järjestelmärekisteriin asennuksen aikana. Toiminnolla varmistetaan, että Lisää/poista sovellus löytää oikean asennuksenpurkuohjelman sen nimestä riippumatta.

Asennuksen poisto on käyttäjille tärkeä toiminto ja se on myös yksi edellytys Windows-yhteensopiva-logon saamiselle. Poistamiseen kuuluu sovelluksen tiedostojen poistaminen (mutta ei dokumenttien) ja järjestelmärekisteriin tehtyjen lisäysten poistaminen.

Koska useat sovellukset jakavat resursseja ja tekevät muutoksia eri puolille järjestelmää, sovelluksen poistamiseen ei yleensä riitä pelkkä tiedostojen tuhoaminen yhdestä alikansioista. Käyttäjien täytyy voida poistaa sovellus täy-

dellisesti, korjata ongelmia uudelleen asentamalla, muuttaa asetuksia ja päivittää sovellusta. Uninstall-toiminto antaa käyttäjälle myös mahdollisuuden vapauttaa levytilaa ja noudattaa lisenssisopimuksia poistamalla sovellus ensin yhdeltä tietokoneelta ja asentamalla sen sitten toiselle.

Windows-sovelluksen poistaminen täytyy suorittaa huolellisesti. Mahdollisia ongelmia syntyy, kun poistetaan sovelluksesta riippuvia tiedostoja kuten DLL- ja ActiveX-kontrolleja. (Riippuvuuksista on kerrottu luvussa 13 oppitunnilla 5.) Uninstall ei saa poistaa moduleja, joista muut sovellukset ovat riippuvaisia.

Sovelluksen poisto-ohjelma ei voi suoraan nähdä, käyttävätkö muut sovellukset jotain modulia, mutta se voi päätellä tämän tutkimalla käyttötilastoa, joka on talletettuna rekisterissä. Käyttötilastosta kerrotaan seuraavassa jaksossa.

Tietojen lisääminen ja poistaminen rekisteristä

Asennusohjelma lisää sovelluksesta tarvittavat tiedot rekisteriin. Käyttäjäkohdattaiset tiedot tulisi kirjoittaa rekisterin

HKEY_CURRENT_USER\SOFTWARE avaimen. Windowsin aiemmissa versioissa nämä tiedot kirjoitettiin Win.ini-tiedostoon. Sovellukseen liittyvät tiedot puolestaan lisätään **HKEY_LOCAL_MACHINE\SOFTWARE**-avaimen alle seuraavassa muodossa:

```
HKEY_LOCAL_MACHINE\SOFTWARE\YrityksenNimi\TuotteenNimi\Versio
```

Kirjoita viistottujen nimien kohdalle sovelluksesi tietoja vastaavat nimet.

PATH-ympäristömuuttuja

Jokaisella käynnistetyllä sovelluksella on oma PATH-ympäristöasetus, joka sisältää luettelon eri kansioden poluista. Luettelo määrittelee polut, joista käyttöjärjestelmän tulee hakea sovelluksen vaatimia suoritettavia moduleita. Windows etsii ensin automaattisesti sovelluksen lataamiseen tarvittavia tiedostoja tietyistä järjestelmäkansioista. Koska useimmat sovellukset asennetaan omiin kansioihinsa, täytyy sovelluksen komponenttien sijainti määritellä sovelluskohtaisella PATH-asetuksella.

Mietitään esimerkiksi MyApp-kansiossa olevaa sovellusta, joka käyttää MyDLL-kansiossa olevia DLL-tiedostoja. Olettaen, että käytetään tavanomaista linkitystä — eli oletetaan, että sovellus ei lataa DLL:iä kutusumalla **LoadLibrary()** API-funktiota — sovellus ei voi toimia ilman, että MyDLL on merkittynä sen PATH-luetteloon. Käyttöjärjestelmä hakee DLL-tiedostoja vain oletuksena olevista järjestelmäkansioista ja kansioista, jotka on määritelty PATH-muuttujassa. Jos tarvittavia DLL-tiedostoja ei löydy, järjestelmä kieltäytyy suorittamasta sovellusta. Sovelluksen asennusohjelman täytyy sen vuoksi määritellä PATH-asetus, joka sisältää MyDLL-kansion, johon vaaditut DLL:t on sijoitettu.

Asennusohjelma rekisteröi PATH-muuttujan kirjoittamalla halutun arvon rekisteriin **HKEY_LOCAL_MACHINE** -juureen **\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths** -avaimeen. Tämä sama merkkijono sisältyi Regstr.h-tiedostossa määriteltyyn **REGSTR_PATH_APPPATHS**-makroon. Huomaa, että viimeinen avain, **App Paths**, koostuu kahdesta sanasta.

Asennusohjelman tulee luoda uusi avain, jolla on sama nimi kuin sovelluksen exe-tiedostolla ja sen jälkeen asettaa Path-arvo, joka sisältää halutun polun. Tässä on esimerkki siitä, miltä MyApp-kansioon asennetun NewApp-sovelluksen Path-arvo saattaisi näyttää rekisterissä:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
App Paths\NewApp.exe
```

```
Default=D:\MyPrograms\MyApp\NewApp.exe
```

```
Path=D:\MyPrograms\MyDLL;D:\MyPrograms\MyApp\Utilities
```

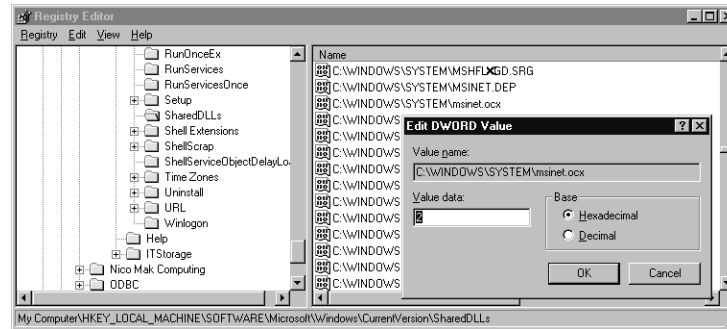
Default-arvo määrittää exe-tiedoston polun kokonaisuudessaan. Käyttöjärjestelmä käyttää tätä arvoa käyttäjän kirjoittaessa sovelluksen nimen **Suorita**-dialogiin määrittelemättä sen polkua. Windows etsii halutun ohjelman käymällä läpi **App Paths** -avaimen ja lukemalla sitten koko polun Default-arvosta.

Path-arvo sisältää NewApp-sovelluksen PATH-asetuksen, jossa on viittaus MyDLL-kansioon. Kun käyttöjärjestelmä lataa NewApp-sovelluksen, se etsii sovelluksen tarvitsemia DLL-tiedostoja ennalta määrätyistä paikoista. Lisäämällä MyDLL-kansion sovelluksen PATH-listaan, asennusohjelma varmistaa, että järjestelmä voi aina löytää tarvittavat DLL:t käynnistäessään NewApp-sovelluksen.

Jaettujen moduulien käyttötilastot

Asennusohjelman tulisi lisätä rekisteriin käyttötilasto jokaiselle asentamalleen moduulille. Kun sovellus poistetaan, uninstall-ohjelma pienentää käyttötilaston arvoa. Jos arvo putoaa nolleen, uninstall voi poistaa modulin — yleensä tosin kysytään ensin käyttäjältä lupa poistamiseen. Jos kaikki asennusohjelmat noudattaisivat tätä käytäntöä, kaikki moduulit voisi poistaa huoletta. Valitettavasti kaikki eivät tee asennusohjelmiaan tämän käytännön mukaan, joten on parasta kysyä käyttäjältä lupa tiedon poistamiseen, vaikka sen käyttötilasto olisikin pudonnut nolleen.

Jaettujen moduulien käyttötilastot talletetaan **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDLLs**-avaimeen. Kuvassa 14.1 seuraavalla sivulla on esimerkki, jossa RegEdit-apuohjelman (käsitellään myöhemmin) avulla nähdään, että msinet.ocx-nimisen ActiveX-kontrollin käyttötilaston arvo on tällä hetkellä 2.



Kuva 14.1 Käyttötilasto RegEdit-apuohjelmalla nähtynä

Cabinet-tiedostot

Cabinet-tiedostossa (cabinet file) tiedostot ovat pakattuina yhdeksi tiedostoksi, joka toimii tiedostokirjastona. Cabinet-tiedostoja, jotka tunnistaa .cab-tarkentimesta, käytetään usein suurten sovellusten jakelussa, koska ne pienentävät tarvittavien levykkeiden tai romppujen määrää. Sovelluksen asennusohjelma lukee .cab-tiedostoja, purkaa niiden sisällön takaisin alkuperäisiksi tiedostoiksi ja siirtää tiedostot käyttäjän kovalevylle.

.cab-tiedosto on samankaltainen kuin .zip-tiedosto, joka toimii arkistona joukolle tiedostoja. .cab-tiedosto voi sisältää millaisia tiedostoja tahansa ja siihen voidaan liittää digitaalinen allekirjoitus, joka yksilöi tiedoston tekijän ja varmistaa, ettei tiedostoa ole vahingoitettu tarkoituksellisesti tai vahingossa. Digitaalisia allekirjoituksia käsitellään tarkemmin tämän luvun oppitunnilla 4.

.cab-tiedoston tekeminen vaatii joko MakeCab tai CabArc (Cabinet Archiver) -apuohjelman. Molemmat kuuluvat Microsoft Cabinet Software Development Kit (SDK) -pakettiin, joka on maksutta haettavissa osoitteesta msdn.microsoft.com/workshop/management/cab/cabdl.asp. MakeCab luo pakattuja imageja, jotka sisältävät tuotteeseen kuuluvat tiedostot ja se on suunniteltu toimimaan yhteistyössä Setup-ohjelmien kanssa. CabArc on konsolipohjainen ohjelma, joka osaa lukea ja kirjoittaa .cab-tiedostoja. CabArc ohjelma tarvitsee parametreikseen .cab-tiedoston nimen ja pakattavien tai purettavien tiedostojen luettelon. Esimerkiksi rivi

```
cabarc n images.cab \myapp\images\*.jpg \myapp\images\*.gif
```

luo Images.cab-tiedoston, ja lisää siihen pakatussa muodossa kaikki MyApp \Images-kansiossa olevat .jpg ja .gif muotoiset grafiikkatiedostot. Valitsin *n* käskää CabArc-ohjelmaa luomaan uuden tiedoston. Cabinet SDK sisältää CabArc ja MakeCab -ohjelmien dokumentaation ja esimerkkejä.

Rekisteritiedostot

Asennusohjelmat hyödyntävät joskus rekisteritiedostoja tehdessään merkintöjä järjestelmärekisteriin. Rekisteritiedostot tunnistaa .reg-tarkentimesta. Rekisteritiedosto on skripti, jossa luetellaan avaimet, arvot ja sijainnit, jotka lisätään rekisteriin. Koska rekisteritiedostot ovat ASCII-muodossa, niitä voidaan katsella ja muokata tekstieditorilla. Esimerkiksi aiemmin mainitun NewApp-ohjelman rekisteritiedosto määrittelee ohjelman PATH-ympäristön kuten seuraavassa koodissa nähdään:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\  
App Paths\NewApp.exe]
```

```
@="D:\\MyPrograms\\MyApp\\NewApp.exe"
```

```
"Path"="D:\\MyPrograms\\MyDLL;D:\\MyPrograms\\MyApp\\Utilities"
```

Vaikka pisimmät rivit tässä on jaettu niin, että ne sopivat painetulle sivulle, täytyy jokaisen merkinnän olla yhdellä katkeamattomalla rekisteritiedoston rivillä. Rekisteriskripteissä ei voida käyttää riviinjatkomerkkejä.

Esitetyssä muodossaan koodi olettaa, että NewApp-ohjelma on sijoitettu MyPrograms\\MyApp -alikansioon, ja osoittaa näin rekisteritiedostoissa piilevän potentiaalisen ongelman. Asennusohjelman täytyy varmistaa, että kaikki rekisteritiedostossa määritellyt polut vastaavat käyttäjän sovellukselle määrittämää sijaintia. Asennusohjelman täytyy näin ollen pystyä muuttamaan .reg-tiedostoa ennen kuin sitä käytetään.

Windows sisältää RegEdit-apuohjelman rekisteritiedostojen lukemista ja kirjoittamista varten. Asennusohjelma voi käyttää RegEditiä skriptien kirjoittamiseen (tai tuomiseen, *import*) käyttäjän systeimirikisteriin seuraavanlaisella komenolla:

```
regedit newapp.reg
```

Tämä komento kirjoittaa NewApp.reg-tiedoston sisällön rekisteriin. Seuraava koodi esittää, kuinka luodaan IEpath.reg niminen tiedosto, joka sisältää kopion Internet Explorerin polkumäärittämisestä:

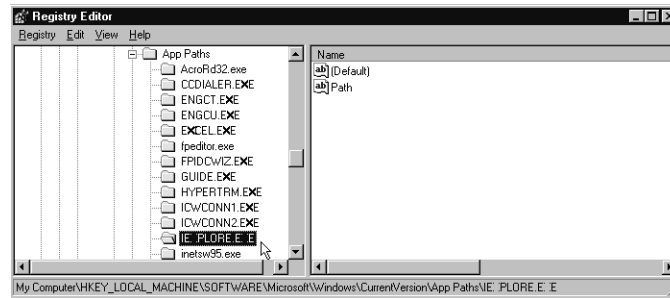
```
regedit /e IEpath.reg "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\  
CurrentVersion\App Paths\IEEXPLORE.exe"
```


/e-valitsin kehottaa RegEditiä lukemaan (tai tuomaan *export*) rekisterin osan määriteltyn tiedostoon. Rekisteriavainta ei ole välttämätöntä sijoittaa lainausmerkkeihin, jos se ei sisällä välilyöntejä.

RegEdit-apuohjelmalla on yksinkertainen Windows-käyttöliittymä, joten sitä ei tarvitse suorittaa komentoriviltä. Seuraavassa harjoituksessa kerrotaan, kuinka kirjoitetaan .reg-tiedosto ohjelman **Registry**-valikon komentoa käyttäen.

► IEpath.reg-tiedoston luominen RegEdit-ohjelmassa

1. Napauta ensin tehtäväpalkin **Käynnistä**-painiketta ja sitten **Suorita**. Kirjoita **Avaa**-ruutuun **regedit**, ja käynnistä sitten RegEdit napauttamalla **OK**.
2. Avaa puunäkymä alkaen **HKEY_LOCAL_MACHINE**-juuresta ja valitse **IEXPLORE.EXE**-avain kuten kuvassa 14.2. **IEXPLORE.EXE**-avaimen polku ilmestyy kokonaan RegEdit-ikkunan alareunaan kuten kuvassa.



Kuva 14.2 Internet Explorerin PATH-merkinnän valitseminen rekisteristä

3. Valitse **Registry**-valikosta **Export Registry File**.
4. Valitse haluttu kansio ja kirjoita **IEpath** kohtaan **File name**.
5. Napauta **Save**, ja sulje RegEdit.
6. Avaa tekstieditori kuten Muistio ja avaa uusi IEpath.reg-tiedosto, joka on sijoitettu oletuskansioon. Skriptistä nähdään, että monien muiden ohjelmien tapaan Internet Explorerin PATH-ympäristöön kuuluu sen oma sovelluskansio.

Reksiteritiedostot ovat hyödyllisiä, kun rekisteriin täytyy tehdä suuria merkintöjä, tai kun tieto on katoavaa ja asennusohjelman täytyy ladata ja poistaa tiedot uudestaan. Asennusohjelma voi myös luoda varmuuskopion halutuista rekisterin tiedostoista RegEditin avulla. Uninstall-ohjelma voi sitten palauttaa alkuperäiset asetukset käyttäjän halutessa poistaa sovelluksen.

Oppitunnin yhteenveto

Tällä oppitunnilla tehtiin yleiskatsaus asennusprosessiin, jossa Setup-ohjelma konfiguroi käyttäjän tietokoneen sovelluksen suorittamista varten.

Asennusohjelman tulisi ideaalilanteessa sisältää asennusvaihtoehdot Compact, Custom, Typical ja Silent. Silent-asennus antaa järjestelmän ylläpitäjälle mahdollisuuden ajaa asennusohjelman ilman ilmoituksia verkon kautta. Ammattilais-tasoinen asennusohjelma pyrkii tekemään asennuksesta käyttäjälle niin helpon kuin mahdollista.

Jos sovellus tarvitsee erityistä PATH-ympäristöä, asennusohjelman tulisi tallettaa haluttu arvo rekisteriin. Se määrittelee PATH-asetuksen, jota Windows käyttää sovellusta suoritettaessa. Asennusohjelman tulisi käsitellä rekisteriä myös asentaessaan jaettuja moduleja kuten DLL-teidostoja ja ActiveX-kontrolleja ja lisätä sinne modulien käyttötilastot. Kun sovellus poistetaan, uninstall-ohjelma pienentää käyttötilaston arvoa ja kysyy käyttäjältä lupaa tiedoston poistamiseen, kun arvo putoaa nolnaan.

Oppitunti päättyy cabinet- ja rekisteritiedostojen kuvaukseen. Cabinet-tiedostot toimivat pakattujen tiedostojen arkistoina. Microsoft tarjoaa käytettäväksi apu-ohjelmia kuten MakeCab ja CabArc cabinet-tiedostojen luomista ja käyttämistä varten. Rekisteritiedostot ovat skriptejä, joita voi liittää järjestelmärekisteriin käyttämällä RegEdit-apuohjelmaa.

Oppitunti 2: InstallShield

Asennus- ja uninstalohjelmien tekemisen helpottamiseksi Visual C++:n Professional ja Enterprise -versiot sisältävät SDK-version Stirling Softwaren InstallShield-ohjelmasta. (Learning ja Standard -versiot eivät sisällä tätä työkalua, mutta voit hankkia sen erillisenä. Vaikka sinulla ei olisikaan työkalua käytössäsi, voit silti hyötyä tämän oppitunnin lukemisesta.)

Tällä oppitunnilla esitellään InstallShield toolkit, jonka avulla voit kirjoittaa komentojonon, joka määrittelee asennuksen yhteydessä suoritettavat toimenpiteet ja vaiheet. InstallShield tekee tämän jälkeen asennusohjelman, joka asentaa sovelluksesi asiakkaan tietokoneeseen.

Kun käytät InstallShieldiä, sinun ei tarvitse kirjoittaa erillistä ohjelmaa asennuksen poistamista varten. InstallShield luo Uninst-nimisen ohjelman, joka lukee asennuksen yhteydessä kirjoitetun lokitiedoston (log file). Lokitiedoston avulla Uninst pystyy perumaan koneeseen asennuksen yhteydessä tehdyt toimenpiteet.

InstallShield tekee projektiasi varten Setup-ohjelman, joka toimii samaan tapaan kuin Setup-ohjelma, jota käytit asentaessasi Visual C++:n. Microsoft tekee useimpiin sovelluksiinsa Setup-ohjelman InstallShieldiä käyttäen.

InstallShieldin yksityiskohtainen käsittely ei kuulu tämän oppitunnin aihepiiriin, vaan tarkoitus on lähinnä esitellä tuotetta. Asennusohjelmat voivat olla nykyään yllättävänkin hienostuneita, pystyen käsittelemään suurta joukkoa erilaisia tiedostotyyppejä, sopeutuen moniin eri käyttöjärjestelmiin ja pystyen käsittelemään monia mahdollisia ongelmatilanteita. Niiden täytyy pystyä toimimaan perinteisten asennusmedioiden, kuten levykkeiden ja romppujen lisäksi myös lähiverkkojen ja Internetin kautta. InstallShield ei ole vaikeakäyttöinen, mutta se sisältää niin suuren määrän asennukseen liittyviä toimintoja, että sen syvällinen oppiminen vie aikaa. Lisäksi tuotteen täydellinen hallinta edellyttää sen skriptikielen, InstallScriptin, opettelemista. Kuten useissa muissakin tapauksissa, kokemus on paras opettaja.

InstallShield käsittelee asennusohjelmaa *projektina*. Jotta sekaannusta tavallisten Visual C++ -projektien kanssa ei syntyisi, tässä luvussa käytetään fraaseja *Setup-projekti* tai *asennusprojekti*, kun viitataan InstallShieldin projekteihin ja *sovel-lusprojektiä* puhuttaessa Visual C++ -tiedostoista, joita asennusohjelma käsittelee asennuksen aikana.

Tämän oppitunnin jälkeen:

- Osaat ottaa InstallShieldin käyttöön Visual C++ -ympäristössä.
- Tunnet InstallShield tutorialin.
- Osaat luoda yksinkertaisen asennusohjelman käyttämällä InstallShieldiä.

Oppitunnin arvioitu kesto: 15 minuuttia

InstallShieldin asentaminen

Jos valitsit InstallShieldin asentaessasi Visual C++:a, Visual C++:n **Tools**-valikossa on jo valmiiksi komento, josta InstallShield Wizard käynnistyy. Jos komentoa ei Visual C++:n **Tools**-valikossa näy, etkä käytä Visual C++:n Learning tai Standard -versiota, sinun tulisi asentaa InstallShield nyt.

► **InstallShieldin asentaminen**

1. Aseta Visual C++ CD-ROM #1 CD-ROM-asemaasi.
2. Käynnistä Ohjauspaneelistä (Control Panel) Lisää/Poista sovellus -toiminto (Add/Remove Programs).
3. Etsi IShield-kansio CD-ROM:lta ja kaksoisnapauta **Setup**.

Setup-ohjelma asentaa InstallShieldin, sijoittaa komennon **Käynnistä/Ohjelmat** (Start/Programs) valikkoon ja lisää **InstallShield Wizard** -komennon Visual C++:n **Tools**-valikkoon.

InstallShield Tutorialin käynnistäminen

InstallShield-ohjelmassa on hyvä opas, joka kertoo, kuinka tutulla Muistio-apuohjelmalla tehdään esimerkki-Setup-ohjelma. Opas käy läpi eri vaiheita luoden lopulta asennusohjelman, joka sijoittaa kuvakkeen ja komennon **Ohjelmat** (Programs) -valikkoon. Kun komento valitaan, se käynnistää Muistion.

Opas näyttää, kuinka InstallShield jakaa sovelluksesi tiedostot eri ryhmiin niin, että voit työstää asennusprojektisi eri osia. Jokaiseen ryhmään kuuluu joukko samantyyllisiä tiedostoja, kuten DLL:iä, ohjelmatiedostoja ja ohjetiedostoja.

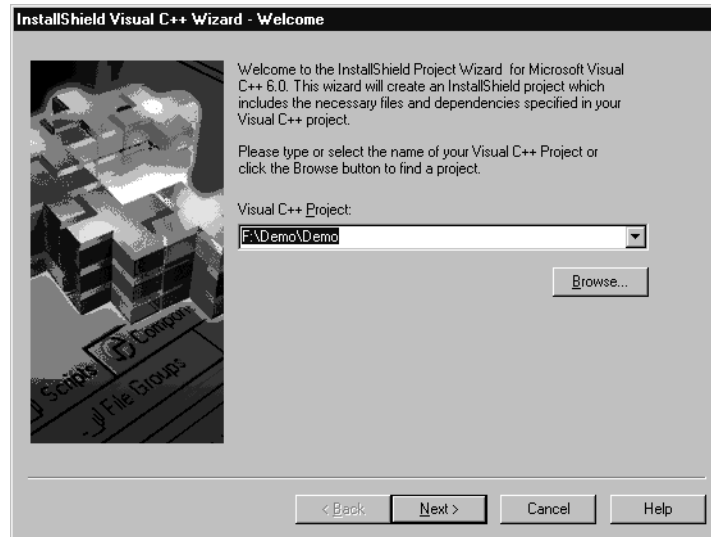
► **InstallShield tutorialin käynnistäminen**

1. Napauta **Käynnistä** (Start), osoita **Ohjelmat** (Programs), ja napauta sitten **InstallShield for Microsoft Visual C++ 6**. InstallShield avautuu.
2. Valitse **Help**-valikosta **Getting Started**.
3. Avaa **Help Topics** -dialogin listasta kaksoisnapauttamalla **Welcome to InstallShield, Tutorials ja Use the Project Wizard**.
4. Kaksoisnapauta ensimmäistä kohtaa **Outline: Use the Project Wizard**.
5. Käy läpi kuusi vaihetta oppaan ohjeita seuraten.

Kun olet käynyt koko Setup-demon läpi, poista se Lisää/poista sovellus-toimintoa käyttämällä kuten oppaan viimeisessä vaiheessa kehoitetaan.

Edellisessä esimerkissä InstallShield käynnistettiin **Ohjelmat** (Programs) -valikosta, jolloin Project Wizard avautuu uuden projektin luomista tai olemassaolevan projektin avaamista varten. Voit käynnistää InstallShieldin Visual C++:sta

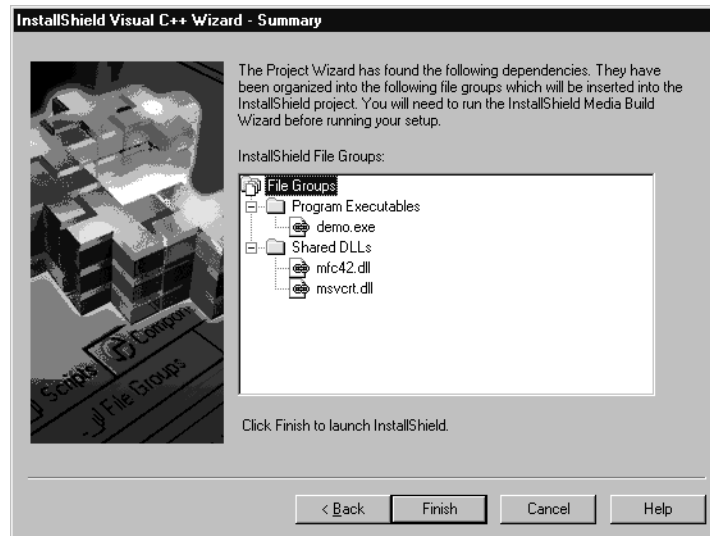
napauttamalla **InstallShield Wizard** -komentoa **Tools**-valikosta. Tämä vaihtoehto on käytännöllinen luotaessa asennusohjelmaa olemassaolevalle Visual C++ -sovellusprojektille, koska silloin siitä saadaan suoraan tietoja, jotka muuten täytyisi syöttää käsin. Kun napautat **InstallShield Wizard** -kometoa, velho näyttää samalta kuin kuvassa 14.3. Valitse haluttu Visual C++ -projekti, napauta **Next**, ja seuraa toiminnon antamia ohjeita.



Kuva 14.3 InstallShield Visual C++ Wizard **Tools**-valikosta käynnistettynä

Ohjelmatiedostojen lisääminen ja poistaminen

InstallShield tutkii sovelluskansion ja paikallistaa kaikki tarvittavat ohjelmatiedostot lisäten ne asennusprojektiin. Kuvasta 14.4 seuraavalla sivulla näet, kuinka toiminto listaa tiedostot, joihin sovellus on yhteydessä. Tässä tapauksessa demo.exe-sovellus on dynaamisesti linkitetty MFC:hen, joten InstallShield lisää MFC42.DLL:n Setup-projektiin. Koska MFC42.DLL puolestaan käyttää C:n run-time-kirjastoa, MSVCRT.DLL-moduli lisätään myös asennusprojektiin.



Kuva 14.4 Lista tiedostoista, joihin sovellus on yhteydessä

Jos aikomuksenesi on jakaa sovellustasi vain paikkoihin, joissa tiedät tiedostojen jo olevan valmiina, et välttämättä halua liittää MFC42.DLL:n ja MSVCRT.DLL:n kaltaisia tiedostoja asennusohjelmaasi. InstallShield Wizard ei anna mahdollisuutta listan muuttamiseen, mutta voit poistaa tiedostot asennusprojektista myöhemmin. Ohjatun toiminnon päätyttyä InstallShield käynnistyy automaattisesti.

Setup.exe-ohjelman luominen

Kun asennusprojekti kääntyy ja toimii moitteettomasti, se täytyy vielä rakentaa lopulliseen muotoonsa. Tässä vaiheessa voit luoda käyttäjille jaettavan Setup.exe-ohjelman ja siihen liittyvät .cab-tiedostot. InstallShield sisältää Media Build Wizard -toiminnon, joka opastaa sinua Setup-ohjelman tekemisessä. Seuraavassa listassa oletetaan, että olet jo luonut asennusprojektin ja olet valmis tekemään siihen Setup-ohjelman.

► Setup-ohjelman luominen

1. Napauta InstallShieldin **Build**-valikon **Media Build Wizard** -komentoa.
2. Kirjoita toiminnon ensimmäisessä vaiheessa (**Media Name**) sovellusprojektin nimi **Media Name** -ruutuun.
3. Toisessa vaiheessa valitaan sovelluksen jakelussa käytettävän asennusmedian koko. Valittavissa on eri kokoisia levykkeitä, CD-ROM, mukautettu koko ja **InstallFromTheWeb** niminen vaihtoehto, joka luo Setup-ohjelman, joka voidaan asentaa Internetin kautta.

4. Valitse kolmannessa vaiheessa joko **Full Build** tai **Quick Build**. **Full Build** vaihtoehto luo jakelussa tarvittavat pakatut tiedostot. **Quick Build** -vaihtoehto on tarkoitettu vain testaukseen ja se antaa mahdollisuuden kokeilla asennusprojektia nopeasti ilman, että pakattuja tiedostoja tehdään. **Quick Build** -toiminnon tekemät levykuvat sisältävät viittaukset sovelluksen tiedostoihin pakattujen tiedostojen sijasta. Tuloksena saatava Setup-ohjelma toimii vain tietokoneella, jossa sovellusprojekti sijaitsee.
5. Toiminnon kolmannessa vaiheessa on **Advanced**-painike. Tätä painiketta napauttamalla saat esiin dialogin, jonka avulla voit tehdä erilaisia valintoja ja asettaa Setup-ohjelmalle salasanan. Voit myös määrätä kansion, johon asennustiedostot kirjoitetaan. Jos et määritä kansiota, käytetään oletuksena kansiota C:\MyInstallations\projekti\Media\projekti, jossa *projekti* edustaa asennusprojektin nimeä.
6. Ohita neljäs vaihe napauttamalla **Next**.
7. Valitse toiminnon viidennessä vaiheessa käyttöjärjestelmä, jossa sovelluksesi voi toimia. InstallShield lisää mukaan tiedostot, joita kyseisissä käyttöjärjestelmissä vaaditaan.
8. Toiminnon kuudes ja viimeinen vaihe on nimeltään **Summary**. Rakenna Setup-ohjelma napauttamalla **Finish**.
9. InstallShield avaa **Building Media** dialogin, jossa edistyminen kuvataan animaation avulla. Kun kaikki on valmista, napauta vain **Finish**-painiketta toistamiseen.

Valitsemassasi asennusprojektikansiossa on nyt Disk Images niminen kansio. Disk Images -kansiossa on vähintään yksi kansio, joka on nimeltään Disk1. Se sisältää asennussarjan ensimmäisen levyn kuvan. Jos asennus vaatii useampia levyjä, niiden kuvat ovat kansioissa Disk2, Disk3 ja niin edelleen. Jokaisen Disk-kansion sisältö täytyy kopioida levykkeille tai muulle asennusmedialle, joka valittiin Media Build Wizardin toisessa vaiheessa. Jos asennus vaatii useamman kuin yhden levyn, täytyy levyihin merkitä levyn numero niin, että käyttäjä osaa vaihtaa levyt oikeassa järjestyksessä asennusohjelman ohjeiden mukaan.

Kokeile vielä ennen sovelluksen toimittamista Setup-ohjelman ajamista toisella tietokoneella, jossa ei ole käänöksessä syntyneitä tiedostoja. Kokeile asennettua sovellusta ja tarkista, että kaikki osat on kopioitu ja rekisteröity oikein ja poista sen jälkeen sovellus varmistaaksesi, että se poistetaan oikein.

Oppitunnin yhteenveto

Tällä oppitunnilla esiteltiin InstallShield-ohjelma, jolla voit tehdä tyylikkäitä asennusohjelmia Visual C++ -projekteille. InstallShield automatisoi paljon Setup-ohjelman luomista etsimällä yhteydet, tekemällä .cab-tiedostot ja levykuvat. InstallShield luo jopa Uninst nimisen asennuksen purkuohjelman.

InstallShield voidaan käynnistää suoraan Visual C++:sta Media Build Wizardin kautta, jolloin se hakee automaattisesti nykyisen sovellusprojektin tiedot. InstallShield auttaa järjestelemään projektin tiedostot erillisiin ryhmiin niiden tyyppin mukaan, jolloin jokaiselle ryhmälle voidaan määritellä erilaisia ominaisuuksia. InstallShield luo asennusta ohjaavan komentojonon käyttäen omaa InstallScript-kieltään. Asennusta voidaan hallita täydellisesti muokkaamalla komentojonoa.

Oppitunti 3: COM-komponenttien rekisteröinti asiakaskoneessa

Asennuksen aikana Setup-ohjelman täytyy rekisteröidä käyttäjän tietokoneeseen kaikki ActiveX-kontrollit ja muut COM-komponentit, joita sovellus tarvitsee toimiakseen. Rekisteriin ActiveX-kontrollista tehtävä merkintä sisältää:

- Kontrollin luokan tunniste GUID.
- Kontrollin suoritettavien tiedostojen sijainti.
- Ominaisuudet, kuten komponentin säiemalli.
- Erilaisia lippuja, jotka määrittelevät kuinka komponentti toimii — esimerkiksi onko komponentti näkyvä vai ei, voidaanko se aktivoida säilön ikkunassa ja niin edelleen.

ActiveX-kontrollit ovat *itserekisteröityviä* (self-registering), eli ne sisältävät omat rekisteröintitietonsa ja ne osaavat itse kirjoittaa tietonsa rekisteriin. Hoidamalla rekisteröinnin itse ActiveX-kontrollit poistavat suuren osan Setup-ohjelman rasituksesta. Tällä oppitunnilla kerrotaan, kuinka InstallShield-projekti saadaan asentamaan ActiveX-kontrollit asiakkaan tietokoneeseen toivotulla tavalla.

Tämän oppitunnin jälkeen:

- Tiedät, mitä ovat monien COM-komponenttien, mukaan lukein ActiveX-kontrollien sisältämät itserekisteröintiominaisuudet.
- Tiedät, kuinka komponenttien julkistamat funktiot kirjoittavat ja poistavat rekisteriin talletettavan COM-informaation.
- Osaat lisätä itserekisteröityviä komponentteja InstallShield-projektiin.

Oppitunnin arvioitu kesto: 15 minuuttia

Itserekisteröityvät komponentit

Setup-ohjelma voi päivittää asennettujen COM-komponenttien tiedot asiakkaan järjestelmärekisteriin käyttämällä tämän luvun oppitunnilla 1 kuvattuja rekisteröintitiedostoja. ActiveX-kontrollit eivät kuitenkaan vaadi rekisteritiedostojen käyttöä, koska ne ovat itserekisteröityviä.

Luvussa 10 kerrottiin, että ActiveX-kontrolli on joko itsenäinen ohjelma tai yleisemmin DLL. Jälkimmäisiä kutsutaan *in-process* (tai *in-proc*) ActiveX-kontrolleiksi. In-process ActiveX -kontrolli julkistaa **DllRegisterServer()**-nimisen funktion, joka kutsuttaessa kirjoittaa rekisteriin kontrollin vaatimat

tiedot. Ainoa ongelma on se, että jonkin toisen sovelluksen täytyy ladata kontrolli ja kutsua sen **DllRegisterServer()**-funktioita.

Tämä tehtävä jää yleensä Web-selaimelle (jos ActiveX-kontrolli ladataan Web-sivulta) tai Setup-ohjelmalle. Koska käyttäjät saavat ActiveX-kontrolleja vain Internetistä tai asennusohjelmien mukana, rekisteröinti tapahtuu näkymättömissä — ja näinollen monien käyttäjien ei koskaan tarvitse huolehtia ActiveX-kontrollien asentamisesta.

Muulla tavoin — esimerkiksi sähköpostilla — saadut komponentit vaativat toimi-akseen rekisteröinnin suorittamista käsin. Windowsissa on apuohjelma RegSvr32 tätä tehtävää varten. RegSvr32 lataa pyydetyn ActiveX-kontrollin tai muun itserekisteröityvän COM-komponentin ja kutsuu komponentin **DllRegisterServer()** -funktioita:

```
regsvr32 path\MyControl.ocx
```

jossa *path* edustaa paikkaa, johon komponentti on talletettu. Komennon:

```
regsvr32 /u path\MyControl.ocx
```

seurauksena kutsutaan komponentin **DllUnregisterServer()**-funktioita, joka poistaa rekisteristä kaikki tiedot, jotka **DllRegisterServer()** kirjoitti. Kommentoon lisätty */u*-valitsin merkitsee asennuksen purkamista.

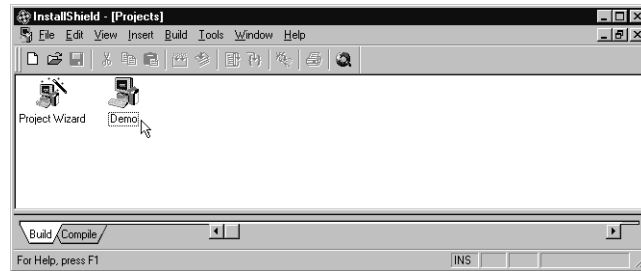
Itserekisteröityvien komponenttien lisääminen InstallShield-projektiin

Jos sovelluksesi tiedostoihin sisältyy ActiveX-kontrolleja tai muita itse-rekisteröityviä komponentteja, luotavan Setup-ohjelman täytyy kutsua komponenttien **DllRegisterServer()**-funktioita asennuksen aikana. Vastaavasti asennusta purkavan ohjelman täytyy kutsua jokaisen poistettavan komponentin **DllUnregisterServer()** -funktioita.

InstallShieldin luomat Setup ja Uninst -ohjelmat pystyvät rekisteröimään ja poistamaan rekisteristä sovellukseesi kuuluvat ActiveX-komponentit. Saadaksesi asennusprojektin käsittelemään itserekisteröityviä komponentteja oikein, sinun täytyy lisätä projektiin tiedostoryhmä, joka sisältää komponenttien tiedostot ja jonka **Self Registered** -ominaisuus on asetettu.

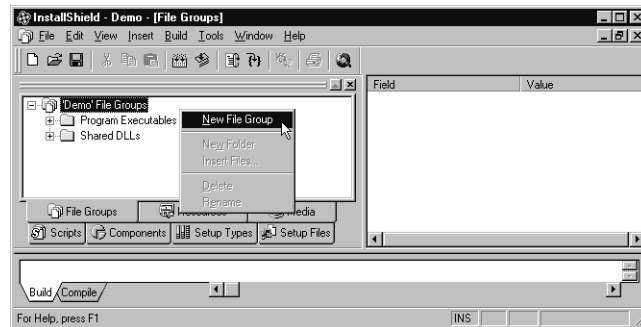
► Ryhmän tekeminen itserekisteröityville komponenteille

1. Avaa asennusprojekti InstallShieldiin kuten kuvassa 14.5.



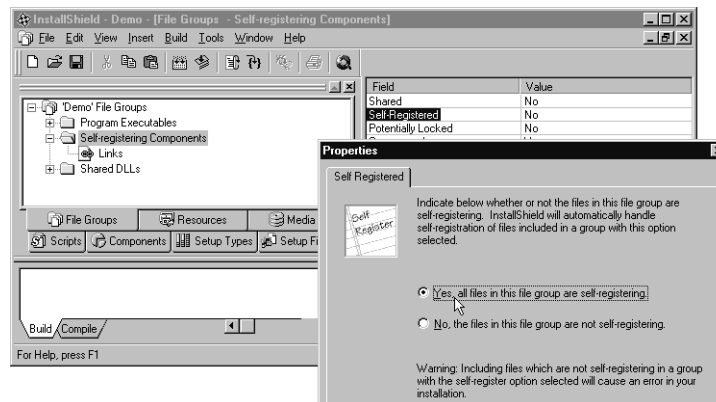
Kuva 14.5 Olemassaolevan projektin avaaminen InstallShieldiin

2. Valitse **File Groups** -välilehti projektin työtilasta ja napauta hiiren oikeanpuoleisella painikkeella File Groups -työtilaa.
3. Valitse pikavalikosta **New File Group** -komento, kuten kuvassa 14.6.



Kuva 14.6 Uuden tiedostoryhmän luominen InstallShieldissä

4. Kirjoita ryhmän nimeksi **Self-registering Components** ja paina ENTER.
5. **Self-registering Components** -ryhmän ollessa työtilassa valittuna kaksoisnapauta työtilan oikeassa reunassa olevaa **Self-Registered** -kenttää.
6. Valitse **Properties**-dialogista **Yes, all files in this group are self-registering**, kuten kuvassa 14.7. Tällä valinnalla kerrotaan, että ryhmässä on vain itserekisteröityviä komponentteja.



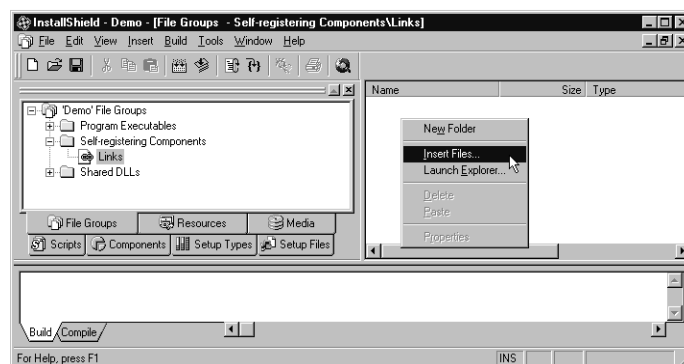
Kuva 14.7 Tiedostoryhmän lisääminen itserekisteröityville komponenteille

7. Napauta **OK**. **Self-Registered**-kentän arvoksi tulee **Yes**.

Kun Setup-ohjelma ajetaan käyttäjän tietokoneessa, se lataa ryhmän jokaisen tiedoston, käynnistää rekisteröinnin kutsumalla sen **DllRegisterServer()**-funktia ja poistaa tiedoston muistista. Seuraavaksi täytyy lisätä varsinaiset komponentitiedostot uuteen tiedostoryhmään.

► **Uusien tiedostojen lisääminen Self-registering Components -ryhmään**

1. Valitse projektin työtilasta **Links**-alataso **Self-registering Components**-kohdan alta.
2. Napauta hiiren oikeanpuoleisella painikkeella missä tahansa työtilan oikealla puolella. Valitse kuvassa 14.8 näkyvästä pikavalikosta **Insert Files** -komento.



Kuva 14.8 Tiedoston lisääminen ryhmään

3. Etsi **Insert file links** -dialogissa ensimmäinen ActiveX-kontrolli tai muu itse-rekisteröityvä kontrolli, jonka haluat lisätä asennusprojektiin ja napauta **Open**.
4. Suorita samat toimenpiteet kaikille komponenteille, jotka haluaa lisätä ryhmään.

Oppitunnin yhteenveto

ActiveX-kontrolli sisältää omat rekisteröintitietonsa, joihin kuuluvat:

- Kontrollin luokan tunniste.
- Kontrollin suoritettavien tiedostojen sijainti.
- Suorituksen aikaiset piirteet ja ominaisuudet.
- Sekalaisia tietoja säilöä varten.

Jokaisella ActiveX-kontrollilla on **DllRegisterServer()**-niminen funktio, joka kutsuttaessa kirjoittaa tiedot järjestelmärekisteriin. ActiveX-kontrollilla on myös vastakkainen **DllUnregisterServer()**-niminen funktio, joka poistaa tiedot rekisteristä. Nämä funktiot tekevät ActiveX-kontrollista itserekisteröityvän ja itse rekisteröintinsä poistavan.

Käyttäjät saavat ActiveX-kontrolleja yleensä katsellessaan Web-sivuja ja asentamalla säilösovelluksia, jotka käyttävät kontrolleja. Molemmissa tapauksissa rekisteröinti tapahtuu näkymättömissä selaimen tai Setup-ohjelman kutsuessa kontrollin **DllRegisterServer()**-funktioita. Käyttäjät voivat itse lisätä ja poistaa ActiveX-kontrolleja käyttämällä Windowsin RegSvr32-apuohjelmaa.

Säilösovelluksen InstallShield-projektiin sisältyy kaksi lisävaihetta. Ensiksi projektiin täytyy lisätä tiedostoryhmä, jonka Self-Registering-lipun arvoksi asetetaan **Yes**. Toiseksi kaikki sovellukseen kuuluvat ActiveX-kontrollit ja muut itserekisteröityvät komponentit täytyy lisätä käsin tähän uuteen tiedostoryhmään. Toinen vaihe on tarpeen, koska ActiveX-kontrolleja ei luetella säilön otsikkotiedostoissa tavallisten DLL:ien tapaan, joten InstallShield ei tunnista niitä riippuvuuksiksi ennen kuin sitä nimenomaan pyydetään niin tekemään. Kun tuloksena saatu Setup-ohjelma ajetaan, se kutsuu jokaisen projektiin kuuluvan itserekisteröityvän komponentin **DllRegisterServer()**-funktioita.

Oppitunti 4: Jakeluvaihtoehdot

Kun Setup-ohjelma on tehty ja se toimii oikein, on sovellus valmiina jakeluun. Tällä oppitunnilla pohditaan, kuinka saat sovelluksesi parhaiten toimitettua käyttäjille ja nostetaan esiin muutamia asioita, joita sinun tulisi pohtia sovellusta julkistettaessa.

Tällä hetkellä kolme tavallisinta Windows-ohjelmien jakelukanavaa ovat:

- Asennusmedia kuten levyke tai CD-ROM.
- Lähiverkko.
- Internet.

Tämän oppitunnin jälkeen:

- Tiedät, mitä etuja ja heikkouksia eri jakelumenetelmillä (medialla, lähiverkolla ja Internetillä) on.
- Tiedät, mitä eroja asennusohjelmissa on toimitettaessa ActiveX-kontrolleja tuotteena muille ohjelmoijille ja toimitettaessa kontrolleja säilöohjelmien mukana käyttäjille.
- Tiedät, mikä on digitaalinen allekirjoitus ja mikä on sen merkitys.
- Tiedät, kuinka kontrolleja lisensoidaan ja kuinka lisensointi vaikuttaa asennusprosessiin.
- Osaat suunnitella World Wide Webiin tarkoitettuja ActiveX-kontrolleja.

Oppitunnin arvioitu kesto: 20 minuuttia

Mediapohjainen jakelu

Levykkeet ja CD-ROM ovat kustannustehokas ja mukava jakelutapa. 3.5-tuumaiset levykkeet ovat lisäksi tasapuolinen jakelutapa, koska lähes kaikilla on käytössään levykeasema, jolla mediaa voidaan lukea.

Levykkeitä käytetään usein sovellusten päivitysversioiden jakeluun yksinkertaisista bugi-korjauksista kokonaan uusiin versioihin saakka. Päivityspakkauksen sisällä tulisi olla Setup-ohjelma, joka asentaa vain päivityksessä muuttuneet tiedostot, tehden näin asennuksesta mahdollisimman nopean ja vaivattoman.

Verkkopohjainen jakelu

Jakelu verkon kautta on halvin vaihtoehto ja on ideaalinen, jos sovelluksesi on tarkoitettu vain yhden yrityksen tai yritysryhmän käyttöön. Tällaisissa tapauksissa käyttäjät voivat muodostaa yhteyden lähiverkon tai laajaverkon kautta. Levykuvat voidaan laittaa verkon kautta jakoon niille käyttäjille, joille sovellus on

tarkoitettu, jolloin he voivat yksinkertaisesti käynnistää Setup-ohjelman verkon kautta.

Palvelinkohtaista lisenssiä käyttävien sovellusten jakelu tapahtuu kahdessa vaiheessa. Ensin organisaatiolle toimitetaan yksi asennuspaketti, josta asennustiedot kopioidaan palvelimelle. Tämän jälkeen lisenssin mukaiset käyttäjät voivat asentaa sovelluksen verkon kautta.

Internet-pohjainen jakelu

World Wide Webin kautta tapahtuva jakelu on yleistynyt viime vuosina. Internet-pohjaisesta jakelusta saadaan useita etuja:

- Ei pakkauskustannuksia.
- Ei jakelukustannuksia normaalin Internet-palvelun ylläpidon lisäksi.
- Ei jälleenmyyjistä koituvia kustannuksia, koska ohjelmiston tekijä toimii myös myyjänä.
- Kansainvälinen myynti tehostuu, koska muista maista tulevat asiakkaat voivat käyttää omalla kielellään toteutettuja Web-sivuja ja ladata vain omaan kieliversioonsa kuuluvat tiedostot.
- Mahdollisuus kokeiluversioiden jakeluun.

Aikarajoitettujen kokeiluversioiden jakelu Internetissä on hyvin tavallista. Tässä mallissa käyttäjä lataa itselleen toimivan ohjelman ilmaiseksi ja kokeilee sitä ennalta määrätyn ajan. Kokeiluajan päätyttyä ohjelma kohteliaasti kieltäytyy toimimasta, kunnes käyttäjä ostaa lisenssin. Lisenssin hankinnan jälkeen käyttäjälle toimitetaan usein sähköpostin kautta avainsana, jonka sovellus tunnistaa osoitukseksi rekisteröinnistä. Kun avainsana on syötetty rekisteröintiruutuun, käyttäjä voi jatkaa sovelluksen käyttämistä ilman aikarajoitusta.

Internet on myös loistava keino päivitysten jakeluun. Ohjelmoija voi sijoittaa tiedostot yleiselle palvelimelle, mahdollisesti vain käyttäjien tiedossa olevalla salasanalla suojattuna. Rekisteröidyille käyttäjille voidaan ilmoittaa uusista päivityksistä sähköpostitse tai sovellus voi itsenäisesti aika-ajoin tarkistaa, onko uusia päivityksiä julkaistu. Jos päivitys on saatavilla, käyttäjälle tulisi aina ilmoittaa siitä ja antaa mahdollisuus päivityksen lataamiseen.

Internet-jakelun mahdolliset ongelmatkin tulisi huomioida:

- Käyttäjille aiheutavat haitat, jos he joutuvat lataamaan suuria tiedostoja hitaan yhteyden kautta.
- Asiakaskunta rajoittuu vain käyttäjiin, joilla on Internet-yhteys.
- Tulee huolehtia siitä, että salasanat annetaan vain rekisteröidyille käyttäjille.

ActiveX-kontrollien jakelu

Tämän luvun oppitunnilla 3 kerrottiin, että ActiveX-kontrollit täytyy rekisteröidä asianmukaisesti käyttäjän järjestelmään, mutta tuote, joka on ActiveX-kontrolli ja tuote, joka on ActiveX-kontrolleja käyttävä säilösovellus, eroavat jake-luvaiheessa toisistaan. Ero on merkittävä varsinkin, jos ActiveX-kontrolliin lii-tetään rekisteröintiin liittyvää koodia.

Tässä osassa puhutaan asioista, jotka täytyy huomioda jaeltaessa ActiveX-kon-trolleja itsenäisinä tuotteina. Jos tuotteesi on ActiveX-kontrolli (tai kontrolli-paketti), asiakkaasi eivät todennäköisesti ole tavallisia käyttäjiä. Sen sijaan he ovat todennäköisesti ohjelmoijia, jotka tekivät komponentteja sovelluksia tai Web-sivuja, jotka puolestaan toimitetaan loppukäyttäjille.

Kontrollien riippuvuus

MFC:tä käyttävät ActiveX-kontrollit täytyy linkittää dynaamisesti MFC42.dll-kirjastotiedostoon. Tällaisen kontrollin asennusprojektiin täytyy näin olen liittää kyseinen DLL, jos jakelu tapahtuu kohteisiin, joissa sitä ei ole valmiina. Toi-saalta ATL pystyy luomaan itseriittoisia ActiveX-kontrolleja, jotka eivät tarvitse MFC:n tai C:n run-time-kirjastoja. Jos haluat ATL-komponenttiesi olevan itse-näisiä, älä lisää C:n run-time-kirjastoja käyttävää koodia ja käytä käännösvaihees-sa Release MinDependency -asetuksia.

Jos tarkoituksesi on jakaa joukko yhdessä toimimaan suunniteltuja ATL-kontrol-leja, on usein tehokkaampaa kääntää ne käyttäen MinSize-asetuksia MinDependency-asetusten sijasta. MinSize:n valinta pienentää merkittävästi kon-trollien kokoa, mutta tekee ne riippuvaisiksi ATL.dll-tiedostosta, joka on kirjas-totiedosto, jonka Visual C++ asentaa System tai System32 -kansioon. Huolimatta 54-kilotavuisesta kirjastotiedostosta, jakelun koko saattaa pienentyä, kun kaksi tai useampia kontrolleja ryhmitellään ja käännetään käyttämällä Release MinSize -asetuksia.

Digitaalinen allekirjoitus

Selain voidaan määrätä turva-asetuksissaan vaatimaan ladattavilta ohjelmilta, kuten ActiveX-kontrolleilta, digitaalista allekirjoitusta. *Digitaalinen allekirjoitus* (digital signature) antaa mahdollisuuden varmistaa, että:

- Tiedostojen sisältö on pysynyt samana allekirjoitushetkestä lähtien.
- Tiedostot ovat lähtöisin turvallisesta lähteestä.

Digitaalinen allekirjoitus varmistaa, että tiedoston sisältöä ei ole muutettu sen jälkeen kun se ensimmäisen kerran julkaistiin ladattavaksi. Allekirjoitus varmistaa tiedostojen lähteen yksilöimällä tiedostot luoneen laillisen toimittajan. Kun lisäät allekirjoituksen ActiveX-kontrolliin, olet laillinen toimija. Laillista toimijaa voidaan pitää vastuullisena ohjelmiston ladattaessa tai suoritettaessa aiheuttamista tuhoista.

Saat digitaalisen allekirjoituksen hankkimalla todistuksen *rekisteröintiviranomaiselta* (certifying authority). Rekisteröintiviranomainen on yhteisö, joka takaa identiteettisi ja rekisteröimäsi asiat. Rekisteröinti sisältää digitaalisen allekirjoituksesi ja valtakirjasi varmistuksen. Ongelmatilanteissa rekisteröintiviranomaisen todistaa identiteettisi. Salaista avainta käyttäen tehtävä digitaalinen allekirjoitus voidaan vahvistaa käyttämällä julkista avainta.

Lisensoidut ActiveX-kontrollit

Kun jakelet lisensoituja ActiveX-kontrolleja tuotteena, asennusohjelmasi tulisi huolehtia lisenssivaatimuksista automaattisesti. Ohjelmoija-asiakkaasi tarvitsevat lisenssin käyttäessään tuotettasi omissa sovelluksissaan tai Web-sivuilla.

Kun esimerkiksi Control Wizard luo projektin lisensoidulle ActiveX-kontrollille, se generoi License.txt-nimisen tiedoston, joka täytyy sijoittaa kontrollin kanssa samaan kansioon. Jos olet määrännyt ActiveX-kontrollisi tallettamaan lisenssitiedot tiedoston sijasta rekisteriin, Setup-ohjelmasi tulisi rekisteröidä tarvittavat tiedot asennuksen yhteydessä.

Riippumatta siitä, kuinka lisenssisuojaus on toteutettu — tiedostona, rekisterimerkintänä tai muulla tavoin — lisenssin täytyy olla ohjelmoijan järjestelmässä tai kontrolli ei anna lisätä itseään. Tätä suojasta kutsutaan *suunnittelunaikaiseksi lisenssisuojaukseksi* (design-time license protection), joka tarkistetaan aina ohjelmoijan yrittäessä lisätä kontrollia Visual Basicin lomakkeelle tai esimerkiksi Visual C++:n dialogieditorilla tehtyyn dialogimalliin. Jos lisenssi ei ole kunnossa, Visual C++ näyttää seuraavan sivun kuvassa 14.9 esitetyn kaltaisen ilmoituksen ohjelmoijan yrittäessä sijoittaa kontrollia dialogiin.



Kuva 14.9 Visual C++:n virheilmoitus, kun on yritetty lisätä ActiveX-kontrollia ilman suunnittelunaikaista lisenssiä

Suorituksenaikainen lisenssi (Run-time license protection) otetaan käyttöön ohjelmoija-asiakkaan saatua ActiveX-komponenttiasi käyttävän sovelluksensa tai Web-sivunsa valmiiksi. Suorituksen aikana säilösovellus välittää avaimen kontrollille todistaen näin, että lisenssi on ollut kehitystyön aikana olemassa. Kun avain on tarkistettu, komponentti latautuu ja toimii normaalisti. Tästä syystä käyttäjän järjestelmässä ei tarvita lisenssiä sovelluksen suorittamista varten.

ActiveX-kontrollin sovittaminen Internetiin

ActiveX-kontrolli toimii Internetissä parhaiten, kun se ilmestyy nopeasti Web-sivulle, valmiina toimintaan. Tämän vuoksi hyvä ActiveX-kontrolli:

- On mahdollisimman pieni.
- Pystyy lataamaan ominaisuustietonsa asynkronisesti.

ActiveX-kontrollin koon pienentäminen

Kun käyttäjä katselee ensimmäistä kertaa Web-sivua, jossa kontrolli esiintyy, käyttäjän selaimen täytyy ladata kontrollin tiedostot. Kontrollin tulisikin olla mahdollisimman pieni, jotta latausaika pysyisi lyhyenä. Kun koodaat ActiveX-kontrollia, pidä erityisesti huolta tiedoston koon pienentämisestä niin paljon kuin mahdollista.

ActiveX-kontrollia kirjoitettaessa tulisi kokonaan välttää C:n run-time kirjaston käyttöä. Näin voidaan pienentää kontrollin tiedostokokoa ja varmistetaan samalla, ettei MSVCRT.DLL-tiedostoa ladata kontrollin mukana. Seuraavassa muutamia vihjeitä ActiveX-kontrollin kirjoittamiseen ilman C:n run-time-kirjastoa:

- Älä käytä C-kirjastokutsuja koodissasi — käytä käyttöjärjestelmän tarjoamia vaihtoehtoisia tapoja. Korvaa esimerkiksi **strcpy()** ja **strcat()** -kutsut vastaavien API-funktioiden kutsuilla **lstrcpy()** ja **lstrcat()**.
- Älä tee jäsenfunktioista static-tyyppisiä, sillä staattiset jäsenfunktiot vaativat run-time-kirjaston alustusta varten.
- Jos käytät ATL:ää kontrollin tekemiseen, varmista, että kääntäjä esimäärittelee **_ATL_MIN_CRT**:n. ATL:n COM AppWizard asettaa määrittelyn automaattisesti; jos olet luonut projektin ilman sitä, lisää

määrittely itse. Ota **Project Settings** -dialogista esiin C++-välilehti ja lisää **_ATL_MIN_CRT Preprocessor Definitions** -kenttään.

Tietojen lataaminen asynkronisesti

Jos kontrolli lataa tietonsa asynkronisesti, kontrolli voi toimia, vaikka selain jatkaa edelleen sen ominaisuuksien lataamista taustalla. Tämä mahdollistaa kontrollin käyttämisen jopa ennen kuin kaikki sen tiedot ovat saatavilla. Jos ActiveX-kontrolli tarvitsee suurta tietomäärää, se tulisi toteuttaa niin, että se ilmestyy Web-sivulle toimintavalmiina tietojoukon olessa vielä vajaa.

Esimerkiksi kontrollin, jossa näytetään videoleike, tulisi aloittaa toimintansa jo ennen koko leikkeen latautumista. Sen sijaan tietojen näyttäminen tulisi aloittaa saatavilla olevilla tiedoilla samalla, kun työsiäettä käytetään jäljellä olevien tietojen puskurointiin sitä mukaa, kun selain ottaa sitä vastaan.

Jakelun tarkistuslista

Tässä osassa on lyhyt tarkistuslista Windows-sovelluksen jakelussa huomioon otettavista sekalaisista asioista:

- Varmista, että jaeltava sovellus on käännetty käyttäen jakeluasetusta, ei debug-asetusta. Vaikka Visual C++ oletuksena optimoi jakeluversion nopeuden, useimmat ohjelmoijat pitävät parempana optiomointia koon suhteen. Näin on varsinkin jaeltaessa ActiveX-kontrolleja.
- Ennen kuin liität riippuvuuksia sovellukseesi varmista, että voit jaella niitä laillisesti. Microsoft sallii sinun jakaa sekä MFC- että C run-time kirjasto-moduleja (MFC42.dll ja MSVCRT.dll) Visual C++ -sovellustesi mukana, mutta muut toimittajat voivat asettaa rajoituksia tuotteidensa jakelulle.
- Käyttäjät eivät tarvitse lisenssejä ActiveX-kontrolleille, joten älä liitä lisenssitietoja säilöohjelmiesi asennuspaketteihin.
- Varmista, että sovelluksen ohjetiedostot ovat mukana asennuspaketissa. Windows Help Viewer WinHlp32.exe:n jakelu ohjetiedostojen mukana ei ole tarpeen, koska Windows-järjestelmissä se on jo asennettuna. Jos sovellus käyttää HTML-ohjetta, tulisi HTMLHELP.dll-tiedosto kuitenkin liittää asennuspakettiin.
- Testaa Setup-ohjelmasi eri vaihtoedot ja niiden toiminta eri käyttöjärjestelmissä. Varmista, että se toimii asianmukaisesti, kun tavataan päällekkäisiä kansioita tai tiedostoja ja kun järjestelmällä on käytössään liian vähän muistia tai levytilaa.
- Tee vastaavat testit asennuksen poisto-ohjelmalle ja varmista, että se poistaa tiedostot ja rekisterimerkinnot oikein.

Oppitunnin yhteenveto

Windows-sovelluksia jaetaan käyttäjille jollain kolmesta menetelmästä:

- Levykkeillä tai CD-ROM:illa.
- Verkon kautta ajettavana Setup-ohjelmana.
- Internetin kautta ladattavan tiedostona.

Tällä oppitunnilla käsiteltiin näitä kolmea menetelmää ja niiden hyviä ja huonoja puolia. Internet-palvelun kautta tapahtuvan jakelun etu ovat pienet jakelukustannukset ja se on ideaalinen jakelutapa kokeiluversioiden jakamiseen.

On tärkeää tietää miten ActiveX-komponenttien jakelu muuttuu, kun niitä jaellaan itsenäisinä tuotteina verrattuna jakamiseen sovelluksen osana. Ero on erityisen merkittävä lisensoitavien ActiveX-komponenttien kohdalla. Lisensoitaviin kontroleihin täytyy liittää rekisteröintitiedot, kun niitä jaetaan ohjelmoijille heidän sovelluksiinsa ja Web-sivuilleen liitettäväksi. Loppukäyttäjät eivät lisensitietoja tarvitse, joten niitä ei pitäisi heille myöskään toimittaa.

Internet-jakeluun tarkoitetut ActiveX-kontrollit tulisi suunnitella niin, että niissä huomioidaan Internetin rajoitukset. Kontrollin tulisi olla mahdollisimman pienikokoinen, jotta latausaika pysyisi lyhyenä ja niiden tulisi ladata suuret tietomäärät taustalla kontrollin ollessa koko ajan käyttäjän käytettävissä. Tietojen lataamista taustalla kutsutaan asynkroniseksi lataamiseksi.

Lopuksi jakelusta tulisi tarkistaa vielä, että:

- Sovellus on käännetty jakeluversioksi.
- Riipuvuustiedosto voidaan jakaa käyttäjille luvallisesti.
- Kaikki liitetiedostot kuten ActiveX-kontrollit, ohje- ja kirjasintiedostot ovat mukana.
- Setup ja uninstall -ohjelmat on testattu ja ne toimivat kuten pitää.

Oppitunti 5: Zero Administration for Windows

Zero Administration for Windows (ZAW) on Microsoftin aloite yritysten verkkoihin liitetystä mikrotietokoneista aiheutuvien kustannusten pienentämiseksi. Aloitteen ytimenä on joukko tekniikoita, joiden tarkoituksena on huomattavasti helpottaa organisaation käyttäjien ohjelmistojen ja laitteistojen hallintaa ja sen tarkoituksena on poistaa tarve ylläpitäjän käynneiltä yksittäisillä verkon työasemilla.

Tällä oppitunnilla tutustutaan ensin ZAW:in yleisellä tasolla. Sen jälkeen keskitytään ZAW:n Windows Installeriin ja Systems Management Serveriin (SMS), jotka ovat kaksi suoraan sovellusten jakeluun liittyvää tekniikkaa.

Tämän oppitunnin jälkeen:

- Osaat luetella ZAW-aloitteen pääkohdat.
- Tunnet Windows Installer -tekniikan perusteet.
- Tunnet SMS:n.

Oppitunnin arvioitu kesto: 15 minuuttia

ZAW:n ominaisuudet

ZAW-ehdotus on osa Microsoft Windows Client Strategy -strategiaa. ZAW on jossain määrin toteutunut Windows NT Server 4:ssä, ja sen on ajateltu olevan tärkeä osa Windows 2000 käyttöjärjestelmää. ZAW:n ominaisuuksia ovat:

- Automaattinen järjestelmän päivitys ja sovellusten asennus.
- Tietojen ja asetusten pysyvä tallentaminen välimuisteihin.
- Keskitetty ylläpito ja järjestelmän lukitseminen.

Automaattinen järjestelmän päivitys ja sovellusten asennus

Sovellusten asentaminen ja ylläpito on tyypillisesti paljon työtä vaativa ja virhealtis prosessi. ZAW yksinkertaistaa tätä prosessia huomattavasti automatisoinnin kautta. Kun uusi järjestelmän osa kuten komponentti, laiteajuri tai käyttöjärjestelmän uusi versio on saatavilla, Windows voi päivittää itsensä automaattisesti uusilla osilla. Järjestelmä voidaan asettaa käynnistymään vähimmäisverkkoasetuksilla, jonka jälkeen se tarkistaa mahdolliset päivitykset Internetistä. Jos päivitys löytyy, järjestelmä voi päivittää itsensä ilman käyttäjän toimia.

Tietojen ja asetusten välittäminen

Tämän luvun aiemmilla oppitunneilla kerrottiin, että asennusohjelman tehtävänä on tallentaa sovelluksen tila- ja konfiguraatiodot asiakkaan rekisteriin. Tämä tallennustapa voi kuitenkin olla ongelmallinen verkkotetussa ja liikuteltavissa ympäristöissä, joissa käyttäjät ovat usein poissa omien työpöytänsä äärestä eivätkä voi käyttää sovelluksiaan ja työkalujaan eri paikoista.

ZAW mahdollistaa paikallisesti tallennettujen tietojen peilaamisen palvelimelle, mikä antaa käyttäjille mahdollisuuden kirjautua verkossa mille tahansa tietokoneelle ja saada aina käyttöönsä oman koneensa asetukset. Tätä tietojen ja asetusten tallennustapaa kutsutaan *tietojen välittämiseksi* (persistent caching).

Keskitetty hallinnointi ja järjestelmän lukitseminen

ZAW sallii verkon ylläpitäjän piilottaa käyttäjien laitteita kuten levyasemia niin, että he näkevät vain yhden kotihakemistoon viittaavan levytunnuksen. Näin käyttäjät eivät pysty asentaamaan laitteita ja ohjelmistoja tietokoneisiinsa oman mielensä mukaan.

Windows Installer

Ehkä kaikkein tärkein ZAW:n sovellusten jakeluun liittyvä osa on Windows Installer, joka automatisoi sovelluksen verkon kautta tapahtuvan asennuksen. Windows Installerilla on monia etuja edellisillä oppitunneilla kuvattuihin perinteisiin asennusmenetelmiin verrattuna:

- Kyky asennetun ohjelman automaattiseen korjaamiseen. Jos esimerkiksi sovellukseen kuuluva tiedosto vahingossa poistetaan, Windows Installer varmistaa, että se korvataan ilman käyttäjän toimenpiteitä sovelluksen käynnistyessä.
- Automaattinen yrityksen laajuinen asennus jopa tuhansille käyttäjille.
- Asennus tarvittaessa, eli COM-komponenttien asentaminen vasta asiakassovelluksen tarvitessa niitä ensimmäistä kertaa. Näin voidaan nopeuttaa asennusta ja välttää kiintolevytilan turhalta varaamiselta komponenteille, joita ei koskaan käytetä.

Windows Installer ei ole ohjelma vaan ennemminkin joukko palveluja. ZAW ei poista Setup-ohjelman tarvetta, eikä muuta sen toimintaa. Windows Installer ei korvaa InstallShieldin tapaisia asennusohjelmia. Pikemmin sellaisten tuotteiden merkitys tulee ZAW:ia käyttettäessä kasvamaan, koska asennuksesta tulee entistä monimutkaisempi toimenpide ja tekniikka muuttuu hienostuneemmaksi. Tulevaisuudessa InstallShield ja muut asennusohjelmat tulevat luomaan Setup-ohjelmia, jotka hyödyntävät Windows Installerin ominaisuuksia ja toimintoja.

Systems Management Server

Systems Management Server (SMS) on ZAW:n palvelu, joka jatkuvasti seuraa verkossa olevia tietokoneita. Kaikkien koneiden löydettyt laitteet ja sovellukset listataan SQL-tietokantaan. SMS:n avulla verkon pääkäyttäjä saa nopeasti ajantasalla olevan yhteenvedon verkon tietokoneista ilman lisätyötä.

SMS tekee sovellusten jakelusta entistä tehokkaampaa, koska se sisältää kyselytoiminnon, jonka avulla asennuspaketti voi käsitellä SMS-tietokantaa. Setup-ohjelma voi näin tutkia jokaisen koneen enne asennusta ja tehdä asennuksen vain niille koneille, jotka täyttävät asetetut vaatimukset. Kun asennusohjelma havaitsee koneen, joka ei täytä vaatimuksia se voi ehdottaa ylläpitäjälle tarvittavia muutoksia kuten laitteistopäivityksiä.

Zero Administration Kit

ZAW:n on ilmoitettu olevan Windows 2000 käyttöjärjestelmän osana, ja monet sen ominaisuudet ovat saatavilla Windows NT 4.0:ään. Jos haluat asentaa ZAW:n Windows NT 4.0:ään, voit ladata Microsoft Zero Administration Kit:n osoitteesta:

<http://www.microsoft.com/windows/zak/zakreqs.htm>

Koko Zero Administration Kit on yksi noin 6 Mt:n kokoinen itsepurkautuva tiedosto. Jos haluat pelkän dokumentaation, se on saatavilla erillisenä tiedostona Microsoft Word -muodossa (415 KB).

Oppitunnin yhteenveto

ZAW on ehdotus, jolla pyritään pienentämään Windows-pohjaisen verkon ylläpidon vaatimia kustannuksia. ZAW:n ominaisuuksiin kuuluvat:

- Sovellusten ja käyttöjärjestelmien automaattinen päivittäminen. Järjestelmä tarkistaa säännöllisin väliajoin tietolähteet kuten Web-sivut ja lataa automaattisesti uudet tiedostot niiden tultua jakeluun.
- Tietojen ja asetusten välittäminen niin, että oman tietokoneen asetukset ja tiedostot seuraavat käyttäjää koneelta toiselle.
- Keskitetty järjestelmän hallinta ja lukitseminen, jonka avulla pääkäyttäjä voi suojata tiettyjä järjestelmiä kätkemällä laitteita käyttäjiltä.

Windows Installer tarjoaa verkon kautta palveluja asennusohjelmille nopeuttaen asennusta ja tehden kiintolevyn käytöstä tehokkaampaa. Koska Windows Installer toimii järjestelmän palveluna, se toimii myös silloin, kun Setup-ohjelma ei ole käynnissä. Koska Windows Installer on aina aktiivinen, se voi korjata sovellusta aina tarvittaessa korvaamalla tiedostoja, jotka on siirretty tai poistettu.

SMS ylläpitää luetteloa verkon tietokoneiden laitteista ja sovelluksista SQL-tietokannassa. Asennusohjelma voi selvittää tietokoneen kokoonpanon SMS tietokannasta ennen asennusta ja suorittaa asennuksen vain niille työasemille, jotka täyttävät määrätty vaatimukset.

Laboratorio 14: STUupload-sovelluksen paketointi ja jakelu

Tässä laboratoriossa STUupload-projekti viimeistellään, tekemällä Setup-niminen asennusohjelma. Kun Setup-ohjelma käynnistetään asiakkaan järjestelmässä, se purkaa .cab-tiedostoihin pakatut tiedostot, kopioi STUupload:n ajettavattiedostot kiintolevylle ja rekisteröi sovelluksen komponentit. Tässä laboratoriossa:

- Luodaan STUuploadin jakeluversio.
- Luodaan InstallShield-projekti asennusohjelmaa varten.
- Lisätään riippuvuudet asennusprojektiin.
- Käännetään ja testataan Setup-ohjelma ja .cab-tiedostot.

Harjoituksessa oletetaan, että asiakaalla, StockWatch Data Services Inc:llä, on laaja tietoverkko, joka yhdistää sen työntekijöitä toisiinsa. Jakelupaketti kopioidaan tästä syystä verkkoon, josta käyttäjät voivat käynnistää Setup-ohjelman ja asentaa STUupload-sovelluksen omille koneilleen.

Kuten tämän luvun oppitunnilla 2 kerrottiin, Visual C++:n Learning ja Standard -versiot eivät sisällä InstallShield-työkalua, mutta sen voi ostaa erillisenä. Jos et voi käyttää InstallShieldiä, lue silti teksti, vaikka et voikaan suorittaa harjoituksia 2 - 4.

Oppitunnin arvioitu kesto: 15 minuuttia

STUupload-sovelluksen julkaisuversion tekeminen

Kun STUupload-sovellus on nyt huolellisesti testattu ja debugattu, voidaan siitä luoda jakeluversio. Jakeluversio on debug-versiota pienempi ja nopeampi, koska kääntäjä ei liitä siihen virheenkorjaustietoja ja koska kääntäjä optimoi koodin pienetäkseen sen kokoa.

► Koon suhteen optimoidun julkaisuversion luominen

1. Avaa STUupload-projekti Visual C++:ssa.
2. Valitse **Build**-työkaluriviltä **Win32 Release** -käännösvaihtoehto, kuten kuvassa 14.10.



Kuva 14.10 Valitse Win32 Release **Build**-työkaluriviltä julkaisukäännöksen tekemistä varten

3. Valitse **Project**-valikosta **Settings**.
4. Napauta **Project Settings** dialogin **C/C++** -välilehteä ja valitse **Minimize Size** vaihtoehto **Optimizations**-ruudusta.
5. Sulje dialogi napauttamalla **OK**.
6. Tee STUupload-sovelluksen julkaisuversio napauttamalla **Build**-työkalurivin **Build**-painiketta.

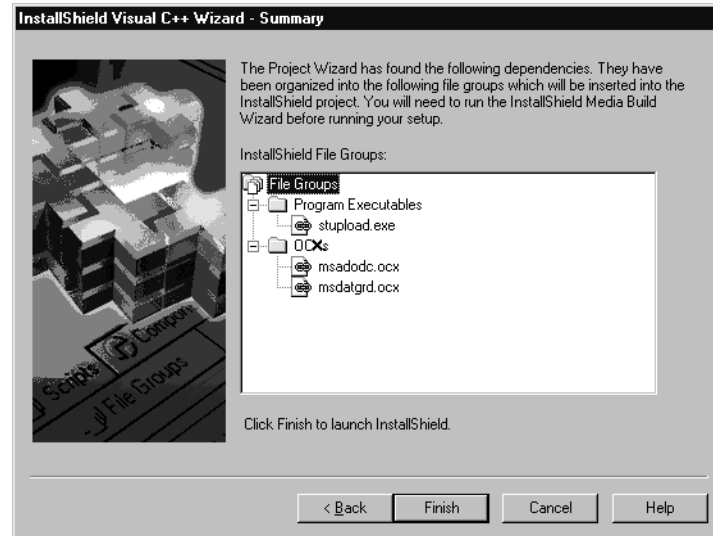
InstallShield-projektin luominen

Nyt kun olet kääntänyt julkaistavan version STUupload-sovelluksesta, voit luoda sitä varten InstallShield-projektin. InstallShield-projekti rakentaa STUupload-sovelluksen ja sen liitetiedostojen jakeluun tarvittavan Setup-ohjelman.

► InstallShield-projektin luominen STUupload-sovellukselle

1. STUupload-projektin olessa edelleen avattuna Visual C++:ssa, napauta **InstallShield Wizardia Tools**-valikosta.
2. Jos STUupload-projektia ei näy toiminnon **Visual C++ Project** -ruudussa, napauta **Browse** ja etsi STUupload.dsw-tiedosto STUupload projektikansioista. Lisää STUupload.dsw **Visual C++ Project** -ruutuun kaksoisnapauttamalla.
3. Siirry seuraavaan vaiheeseen napauttamalla **Next**.
4. Varmista tietojen oikeellisuus ja siirry sitten eteenpäin napauttamalla **Next**.

Toiminto näyttää yhteenvedon STUpload.dsw workspace-tiedostosta kerätyistä tiedoista. Kuten seuraavalla sivulla olevassa kuvassa 14.11, tietoihin sisältyvät kahden luvussa 7 lisätyn ActiveX-kontrollin nimet msadodc.ocx ja msdatgrd.ocx. msadodc on Microsoft ADO Data Control, ja msdatgrd on Microsoft DataGrid -kontrolli. Toiminto lukee kontrollien luokkien tunnisteet, jotka on talletettu STUpload.dsw-tiedostoon ja hakee sitten järjestelmärekisteristä .ocx-tiedostojen tarkan sijainnin.



Kuva 14.11 InstallShield Wizardin Summary-sivu

5. Käynnistä InstallShield napauttamalla **Finish**. Uusi STUpload asennusprojekti avautuu automaattisesti.

Tarvittavien tiedostojen lisääminen projektiin

STUpload-sovellus tarvitsee luvussa 9 lisätyn STLoadData.dll-komponentin palveluja. STLoadData ei ole ActiveX-kontrolli, mutta se on tästä huolimatta itserekisteröityvä, koska sillä on **DllRegisterServer()** ja **DllUnregisterServer()**-funktiot. Komponentti lisätään tästä syystä ryhmään, jonka **Self-Registered**-lippu on **Yes**.

Tarvittava tiedostoryhmä on jo olemassa projektissa, joten uuden ryhmän luominen ei ole tarpeen. InstallShield on luonut ActiveX-kontrolleja varten OCXs-tiedostoryhmän ja asettanut sen **Self-Registered**-lipun. Tässä harjoituksessa lisätään STLoadData-komponentti OCXs-tiedostoryhmään.

► **STLoadData-komponentin lisääminen OCXs-ryhmään**

1. Valitse InstallShieldin workspaceikkunassa **File Groups** -välilehti ja avaa OCXs-ryhmän lista.
2. Napauta hiiren kakkospainikkeella **Links**-merkintää OCXs:n alla ja valitse pika valikosta **Insert Files**.
3. Etsi kansio, jossa STLoadData.dll on, valitse se ja napauta **Open**. OCXs-ryhmän tiedostoryhmään on lisätty STLoadData.dll.

Myös StloadData.dll:n tietokantayhteyden muodostamisessa käyttämä tiedosto täytyy ottaa mukaan.

► **Data Links -tiedostoryhmän ja linnkitiedoston lisääminen**

1. Napauta hiiren kakkospainikkeella **File Groups** -välilehden **STUpload File Groups** -kohtaa. Lisää uusi tiedostoryhmä **Data Links**.
2. Lisää **\DataLinks\STLink.udl**-tiedosto, jonka teit luvun 10 laboratoriossa, **Data Links** -ryhmään.

Huomio Näissä harjoituksissa oletetaan, että Microsoft Data Access Components (MDAC) -komponentit on asennettu asiakkaskoneille. MDAC:n jakelusta saat lisätietoja Visual C++:n ohjeen artikkelista "Redistributing Microsoft Data Access Components".

Asennusohjelman kääntäminen ja testaaminen

Tässä harjoituksessa käytetään InstallShieldin Media Build Wizardia asennusohjelman tekemiseen. Setup-tiedoston lisäksi asennuspakettiin tulee kuulumaan joukko informaatiotiedostoja ja .cab-tiedosto, joka sisältää neljän asiakkaan verkkoon asennettavan ajettavan tiedoston pakatut kopiot. Ajettavat tiedostot ovat STUpload.exe ja sen komponentit MSAdodc.ocx, MSDatGrd.ocx, ja STLoadData.dll.

► **Setup-ohjelman kääntäminen STUpload-sovellukselle**

1. Valitse InstallShieldin **Build**-valikosta **Media Build Wizard**.
2. Kirjoita Media Build Wizardin ensimmäisessä vaiheessa **Media Name** -ruutuun **STUpload**, ja napauta sitten **Next**.

3. Valitse mediatyypeistä 1.44-MB 3.5-inch levyke. Tällä valinnalla määritetään kuinka asennuspaketti välitetään StockWatch Data Servicesille, joka kopioi sen sitten verkkoonsa työntekijöiden käytettäväksi.
4. Ohita Build Type, Tag File, ja Platform -vaiheet napauttamalla neljä kertaa **Next**.
5. Rakenna projekti napauttamalla velhon Summary-vaiheessa painiketta **Finish**.
6. Sulje velho Media Build Wizardin lopetettua napauttamalla **Finish**.

Disk Images -kansiossa C:\MyInstallations\STUpload\Media\STUpload-kansion alla on nyt kansio, jonka nimi on Disk1. Disk1 sisältää kaikki STUpload-sovelluksen asennuksessa tarvittavat tiedostot. Jakelua varten Disk1 kopioidaan yhdelle 3.5-tuuman levykkeelle.

Asennus käynnistetään asettamalla levyke toisen tietokoneen levykeasemaan ja käynnistämällä Setup-ohjelma Lisää/Poista sovellus -toiminnon kautta. Jos Setup-ohjelma kysyy sarjanumeroa, pääset eteenpäin kirjoittamalla mitä tahansa. Kun asennus on päättynyt, varmista, että asennus on onnistunut ja kaikki COM-komponentti on rekisteröity käynnistämällä STUpload-sovellus. Voit poistaa ohjelman ja testata uninstall-ohjelmaa käynnistämällä Lisää/Poista-sovelluksen uudelleen ja valitsemalla poistettavaksi STUpload-sovelluksen.

Kertaus

1. Mitkä ovat jakelun kaksi vaihetta?
2. Mitä ovat .cab- ja rekisteritiedostot?
3. Mitä InstallShield-työkalulla tehdään?
4. Pitääkö ActiveX-kontrollit rekisteröidä järjestelmään enen kuin ne toimivat?
Jos pitää, mitä esimerkkejä voit antaa kontrollin rekisteröintitiedoista?
5. Nimeä kolme sovelluksen jakelutapaa.
6. Mikä on ATL:n käytöstä saatava suurin etu verrattuna MFC:hen ActiveX-kontrolleja tehtäessä?
7. Mitkä ovat kaksi ActiveX-kontrollien lisensoinnissa käytettyä tapaa?
8. Olet tehnyt lisensoitavan ActiveX-kontrollin ja olet myynyt sen kopioin ohjelmistoyhtiölle. Pitääkö sinun antaa heille myös kopio kontrollin lisenssistä? Miksi näin?
9. Nimeä kaksi Internet-sivuilla käytettäväksi suunnitellun ActiveX-kontrollin ominaisuutta.
10. Mikä on Zero Administration for Windows?