

## L U K U 2

# Visual C++ Development Environment

**Oppitunti 1: MFC-projektin luominen 40**

**Oppitunti 2: Ohjelmointiympäristö 49**

**Oppitunti 3: Lähdekoodin hallinta 58**

**Laboratorio 2: STUupload-projektin luominen 69**

**Kertaus 71**

## Tässä luvussa

Tässä luvussa tutustut ohjelmointiympäristöön ja opit työkalut, joiden avulla voit luoda sovelluksia Microsoft Visual C++:lla. Opit, kuinka luot AppWizardia käyttäen kehitysprojektin, joka kokoaa yhteen lähdekoodin ja resurssitiedostot, jotka käännetään suoritettavaksi ohjelmaksi. Tutkit Visual C++-ohjelmointiympäristön toimintoja ja opit, kuinka konfiguroit asetukset projektillesi sopiviksi. Opit myös, kuinka käytät Microsoft Visual SourceSafea sovelluskehitysryhmän tuottaman lähdekoodin hallintaan.

## Ennen kuin aloitat

Tämän luvun oppituntien läpi käyminen edellyttää, että Visual C++ on asennettu luvun 1 oppitunnilla 2 kuvatulla tavalla.

## Oppitunti 1: MFC-projektin luominen

Tällä oppitunnilla esitellään Visual C++ -ohjelmointiprojektit ja kerrotaan, kuinka AppWizard-ohjelmaa käytetään Microsoft Foundation Classes (MFC) -luokkiin perustuvan projektin luomiseen. Käytät AppWizardia sovelluksen rungon koodin ja resurssien luomiseen. Sovellusrunko on mahdollista kääntää ajettavaksi ohjel-maksi. Tätä runkoa voit käyttää pohjana tehdessäsi erikoisvaatimuksesi täyttävää sovellusta.

---

### Tämän oppitunnin jälkeen:

- Tiedät kuinka projektit ja työtila on organisoitu Visual C++:ssa.
- Osaat kertoa millaisia projekteja voit luoda AppWizardilla.
- Tiedät AppWizardin MFC-ohjelman luomiseen liittyvät vaiheet.

**Oppitunnin arvioitu kesto: 30 minuuttia**

---

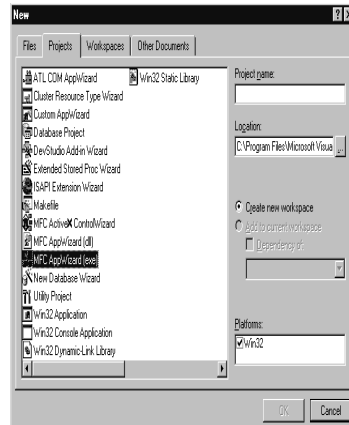
## Projektit, asetukset ja työtilat

Visual Studiossa sovelluksen lähdekooditiedostot ja tiedostot, jotka määrittelevät sovelluksen resurssit kuten valikot, työkalupalkit ja dialogit, on ryhmitelty *projekteiksi*. Projektin avulla voit muokata tiedostojasi ja hallita niiden välisiä suhteita, kuten niiden keskinäisiä riippuvuuksia.

Visual C++ -projektin kehitystyökalujen, kuten kääntäjän, resurssikäntäjän ja linkerin *asetuksia* (configuration), käsitellään keskitetysti Project Settings -dialogissa. Voit määritellä vapaavalintaisen määrän erilaisia asetusten konfiguraatioita projektiisi. Kun käytät AppWizardia projektin luomiseen sekä Debug- että Release-asetukset luodaan automaattisesti.

Projektit sijoitetaan aina *työtiloihin* (workspace). Oletuksena on, että yksi projekti luodaan yhteen työtilaan ja molempien asetuksilla on sama nimi. Organisoidaksesi työtäsi tehokkaammin voit sijoittaa useita toisiinsa liittyviä projekteja samaan työtilaan. Voit myös luoda luottamussuhteita projektien välille varmistaaksesi niiden yhdenmukaisuuden.

Vaikka työtilassa voi olla samanaikaisesti useita projekteja, käsittelet vain yhtä projektia kerrallaan. Käsiteltävää projektia nimitetään *aktiiviseksi projektiksi*. Kuvassa 2.1 on näkymä projektista, jossa on kolme projektia samassa työtilassa. Huomaa, että aktiivinen projekti on merkitty lihavoinnilla.



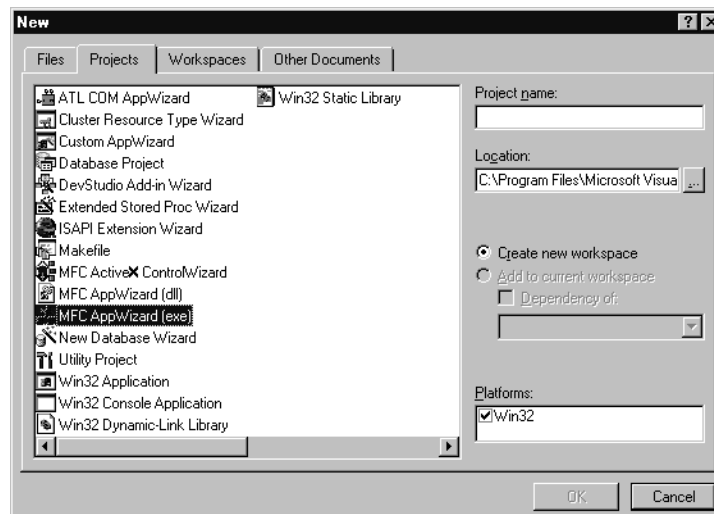
Kuva 2.1 Workspace-ikkuna (FileView)

## Uuden projektin luominen

Kaikki kehitystyö Visual C++:ssa käynnistyy **New Project** -dialogista (kuvassa 2.2), jossa projektityypit ovat lueteltuina.

### ► New-dialogin avaaminen

Käynnistä Visual C++ napauttamalla **Käynnistä** (Start), osoita **Ohjelmat** (Programs), **Microsoft Visual C++ 6.0**, ja napauta sitten **Visual C++**. Valitse **File**-valikosta **New**.



Kuva 2.2 Visual C++:n New-dialogi (Enterprise Edition)

**New**-komentoa käytetään luotaessa uusia työtiloja, projekteja, lähdekooditiedostoja ja resurssitiedostoja. Sillä voidaan tehdä myös muiden Microsoft-sovellusten kuten Microsoft Wordin tai Microsoft Excelin tyhjiä dokumentteja.

## Visual C++:n projektityypit

Visual C++ **New Project** -dialogi antaa sinulle mahdollisuuden luoda uusia sovellusten, komponenttien ja kirjastojen kehitysprojekteja käyttäen seuraavia menetelmiä:

- **MFC development** MFC development -vaihtoehto luo MFC:hen perustuvan kehyksen, jota voit käyttää kehittäessäsi sovelluksia, DLL:iä tai ActiveX-kontrolleja. Muut projektityypit, kuten Internet API:n (ISAPI) Extension Wizardin luomat, perustuvat myös MFC-kehykseen.

Vaikka MFC säästää aikaa ja vaivaa Windows-ohjelmaa luotaessa, sen käyttö pieniä sovelluksia tehtäessä tuottaa turhaa koodia ja heikentää suorituskykyä. Jotta MFC-sovellusta voitaisiin käyttää, täytyy sen koodiin linkittää MFC:n staattiset-kirjastot tai on varmistettava, että MFC DLL:t on asennettu kaikkiin koneisiin, joissa sovellusta käytetään. Pohdi tarvitseeko sovelluksesi MFC:n tarjoamia projektikehityksiä.

- **Win32 development** Win32-vaihtoehtoa käyttäen voit luoda yksinkertaisen Windows-sovelluksen ilman MFC:n tuomaa lisälastia. Voit halutessasi luoda tyhjän projektin, jonka asetukset on sovitettu Windows-sovelluksille, jolloin voit aloittaa Windowsohjelma rakentamisen tyhjästä. Vaihtoehtoisesti voit halutessasi luoda Windows-ohjelman perusrakenteen käyttäen velhoa. Velho huolehtii tarvittavista perustoimista, kuten Windows-luokkien rekisteröinnistä, käyttäjän syöttämiä tietoja ohjelmaan tuovan sanomasilmukan käyttöönnotosta ja käyttöliittymän tuottamien sanomien käsittelystä vastaavien proseduurien toteuttamisesta. Voit luoda myös Windows DLL:ejä. Myös tässä tapauksessa voit lähteä liikkeelle tyhjästä projektista tai luoda projektin, jossa on mukana mallikoodia, jossa näyttetään, kuinka luokkia, funktioita ja muuttujia viedään DLL:istä. Voit myöskin luoda yksinkertaisen konsolisovelluksen, joka suoritetaan komentokehoteesta ja jolla ei ole graafista käyttöliittymää. Lisäksi on mahdollista luoda staattisia kirjastoja, jotka linkitetään ohjelmaan käännösvaiheessa. Nämä vaihtoehdot mahdollistavat myös MFC:en lisäämisen ohjelmiisi. Tätä voit hyödyntää, jos haluat käyttää hyväksesi MFC:n merkkijono- tai kokoelmaluokkia.
- **ATL development** ActiveX Template Library (ATL) on kokoelma mallipohjaisia C++-luokkia, jotka auttavat pienten, nopeiden COM-objektien luomista. Käsittelemme ATL:ää tarkemmin luvuissa 9 - 11. ATL COM App Wizardilla voit luoda COM-palvelimen, DLL:iä tai .exe-tiedostoja, jotka isännöivät COM-komponentteja. Kun olet luonut ATL projektin velhoa käyttämällä, voit lisätä siihen lukuisia ATL pohjaisia COM-objekteja. Niihin voi kuulua yksinkertaisia COM-objekteja, objekteja, joita Microsoft Transaction Service -palvelut tai Active Server Page -sivut voivat hyödyntää, Microsoft Management Console SnapInsejä, ActiveX-käyttöliittymä-kontrolleja ja OLE DB -tiedolähteitä ja tiedon käyttäjiä. Muutkin

projektityypit, kuten DevStudio Add-In Wizard -velhon luomat, perustuvat myös ATL-malleihin.

- **Miscellaneous projects** Koostuu joukosta Visual C++ -toimintoja, jotka eivät sovi edellisiin ryhmiin. Näihin kuuluvat toiminnot, jotka mahdollistavat Microsoft Internet Information Server (IIS) tai Microsoft Cluster Server -palvelimen käyttämien DLL-resurssien luomisen. Voit myös tehdä omia Visual Studio add-inejä, luoda yleiskäyttöisiä yleishyödyllisiä projekteja ja kehittää omia AppWizardeja.

---

**Huomio** Visual C++:n Enterprise-versiossa on joitain projektivaihtoehtoja, joita Standard ja Professional versioissa ei ole. Ne mahdollistavat työskentelyn DEFINITION (ODBC) -tietokantojen kanssa suoraan Visual Studiosta ja SQL Serverin laajennettujen proseduurien luomisen.

---

## AppWizardin käyttäminen

Tässä harjoituksessa luodaan AppWizardilla MFC-projekti.

### ► AppWizardin käynnistäminen

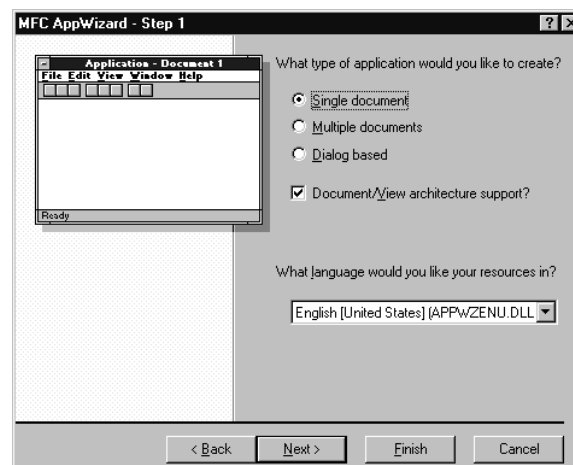
1. Valitse **MFC AppWizard (exe)** -toiminto (kuvassa 2.2).
2. Kirjoita **MyApp Project name** -ruutuun ja napauta OK.

---

**Huomio** AppWizard luo projektin oletuskansioon: C:\Program Files\Microsoft Visual Studio\MyProjects. Halutessasi voit vaihtaa kansiota kirjoittamalla uuden sijainnin **Location**-muokkausruutuun.

---

Kuvassa 2.3 näkyvä ensimmäinen MFC AppWizard -dialogi avautuu.



**Kuva 2.3** MFC AppWizard — Vaihe 1

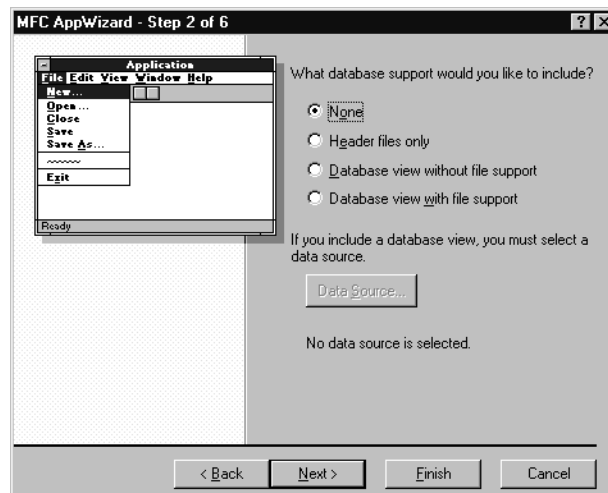
Ensimmäisenä MFC AppWizard antaa sinulle mahdollisuuden valita, onko tarkoituksena luoda yhden dokumentin (SDI) sovellus (**Single document** -vaihtoehto), useamman dokumentin (MDI) sovellus (**Multiple documents** -vaihtoehto), vai sovellus, joka rakentuu dialogi-ikkunan ympärille (**Dialog based** -vaihtoehto). SDI-sovellus sallii vain yhden dokumentin avaamisen kerralla sovelluksen pääikkunaan. MDI-sovellus mahdollistaa useamman dokumentin avaamisen samaan sovelluksen kantaikkunaan erillisiin lapsi-ikkunoihin. Dialogipohjainen sovellus käyttää dialogi-ikkunaa ohjelman pääikkunana tavanomaisen ikkunan sijasta. Luvun 4 oppitunnilla 3 kerrotaan tarkemmin tämän tyyppisistä sovelluksista.

Saat ohjeen mistä tahansa AppWizardin toiminnosta valitsemalla toiminnon ja painamalla F1-näppäintä.

Seuraavat ohjeet kertovat, kuinka tehdään MyApp.exe-niminen SDI-sovellus.

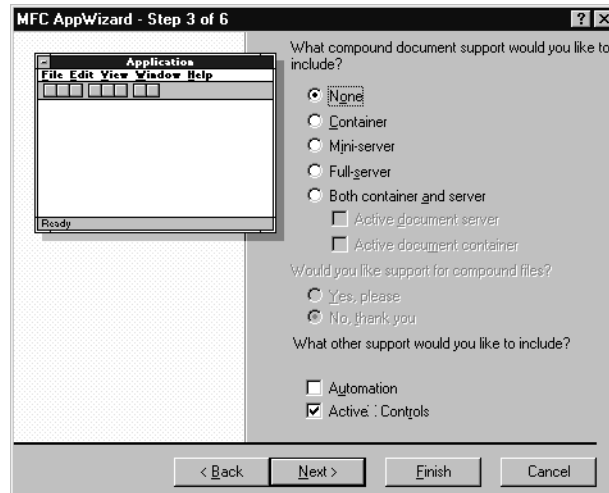
#### ► Projektin luominen AppWizardilla

1. Vaihda sovelluksen tyyppi **Multiple documents** -vaihtoehdosta vaihtoehtoon **Single document**. Siirry seuraavaan dialogiin napauttamalla **Next** (kuvassa 2.4).



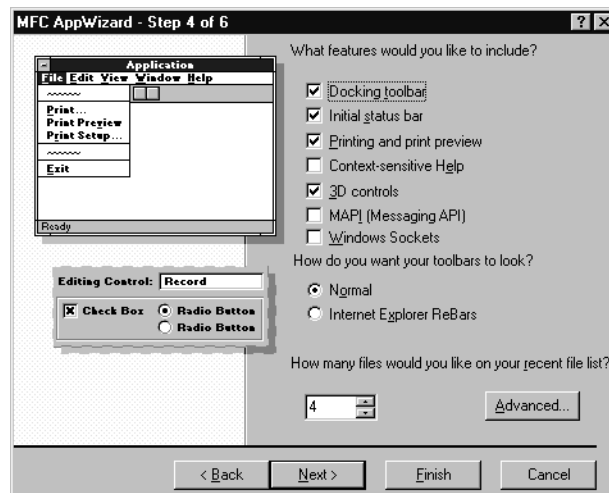
**Kuva 2.4** MFC AppWizard — Vaihe 2

2. MFC:n tietokantatoimintoja käsitellään tarkemmin luvussa 7, *Tietokantatuenn lisääminen*. Anna vaihtoehdon **None** olla valittuna ja siirry eteenpäin painamalla **Next**-painiketta. (Seuraava dialogi on esitetty kuvassa 2.5.)



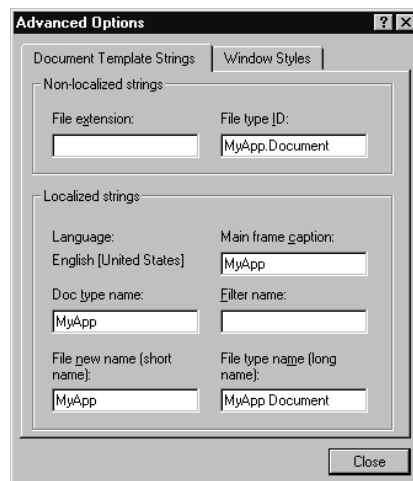
Kuva 2.5 MFC AppWizard — Vaihe 3

3. Tämä dialogi esittää ActiveX-tekniikoihin liittyvät vaihtoehtot. ActiveX-tekniikat käyttävät COMia mahdollistaakseen ohjelmakomponenttien vuorovaikutuksen toistensa kanssa, riippumatta kielestä, jolla ne on tehty. OLE on toinen COMiin perustuva tekniikka. Sitä käytetään sovelluksissa, joilla voidaan käsitellä yhdysdokumenteja — dokumentteja, jotka voivat sisältää useammilla sovelluksilla luotua tietoa. Käsitlemme COMia ja ActiveX-tekniikoita syvemmin luvuissa 8 - 12. Jätä oletusvaihtoehto valituksi ja napauta **Next**-painiketta siirtyäksesi vaiheeseen 4 (kuvassa 2.6).

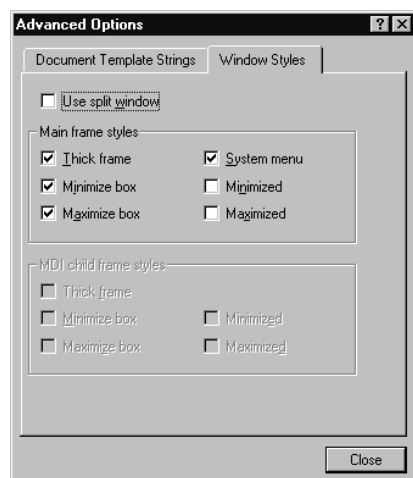


Kuva 2.6 MFC AppWizard — Vaihe 4

4. Näkyvien AppWizardin vilintojen avulla voit määrätä, mitä ominaisuuksia haluat sovelluksesi tukevan. Jätä oletusvaihtoehto valituksi ja napauta **Advanced**-painiketta.
5. **Advanced Options** -dialogissa on kaksi välilehteä. **Document Template Strings** -välilehti (kuvassa 2.7) tarjoaa sinulle mahdollisuuden valita minkä tyyppisiä dokumentteja haluat ohjelmaasi liittää. Tämä on hyödyllistä, koska näin Windows voi yhdistää dokumenttityypin sovellukseen. Käyttäjä voi kaksoisnapauttamalla tiedostonnimeä avata dokumentin sopivaan sovellukseen. **Windows Styles** -välilehti (kuvassa 2.8) antaa mahdollisuuden määrittää sovellusikkunan erityispiirteitä.



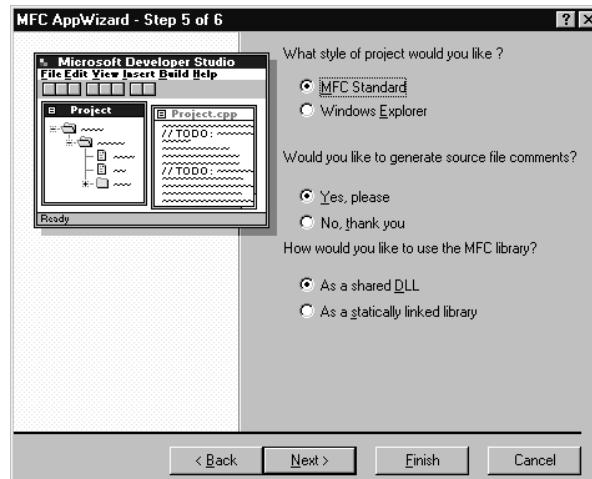
Kuva 2.7 Advanced Options -dialogin Document Template Strings -välilehti



Kuva 2.8 Advanced Options -dialogin Window Styles -välilehti

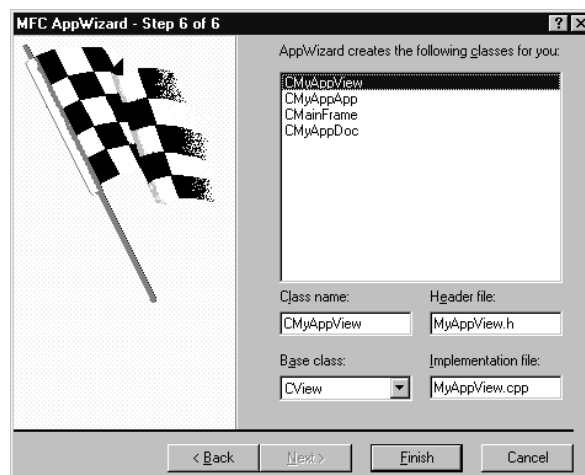


6. Kirjoita **Document Template Strings** -välilehdellä olevaan **File extension** -ruutuun **mya** ja napauta **Close**-painiketta. Siirry seuraavaan dialogiin (kuvassa 2.9) napauttamalla **Next**-painiketta.



Kuva 2.9 MFC AppWizard — Vaihe 5

7. Ensimmäinen valinta tässä dialogissa tarjoaa mahdollisuuden luoda sovelluksen, jonka käyttöliittymä on samanlainen kuin Windowsin Resurssienhallinta-ohjelmalla — pääkohdat esittävä puunäkymä ja luetteloruutu vierekkäin jaettussa ikkunassa. Jätä **MFC Standard** valituksi. Viimeinen valinta määrää, kuinka MFC-kirjastot jaetaan sovellukseksi mukana. Jätä vaihtoehto **As a shared DLL** valituksi, ja siirry AppWizardin viimeiseen dialogiin, jonka näet kuvassa 2.10, painamalla **Next**-painiketta.



Kuva 2.10 MFC AppWizard — Vaihe 6

8. AppWizardin viimeinen dialogi (kuvassa 2.10) kertoo tietoja luokista ja tiedostoista, jotka luodaan valmiiksi puolestasi. Voit muuttaa automaattisesti luotuja nimiä, jos et ole niihin tyytyväinen. Huomion arvoista on, että voit muuttaa kantaluokkaa, josta näkymäluokka johdetaan. Näkymäluokkaa käytetään MFC-sovelluksissa sovelluksen tietojen näyttämiseen. Napauta **Base Class** -alasvetovalikkoa nähdäksesi valittavissa olevat vaihtoehdot.

Tässä valikossa on hyvin vahvoja toimintoja. Moniin lueteltuihin kantaluokkiin sisältyy Windowsin yleisiin kontrolleihin perustuvia toimintoja, joita voit käyttää helpottamaan sovelluksesi tietojen näyttämisen järjestelyä. Jos valitset **CRichEditView**-vaihtoehdon, voit luoda sovelluksen, joka perustuu täysimittaiseen tekstieditoriin. **CHtmlView**-valinnalla luodaan HTML-selaimen perustuva sovellus. Oletuksena oleva **CView** luo sovellukseen tyhjän ikkunan, jossa voit esittää tietoja haluamassasi muodossa. **CScrollView** vastaa **CView**-vaihtoehtoa, mutta se luo vieritettävän näkymän, joka mahdollistaa suuremman tietomäärän esittämisen yhdessä ikkunassa.

9. Valitse **Base Class**-listasta **CScrollView**. Siirry **New Project Information** -dialogiin napauttamalla **Finish**-painiketta. AppWizard esittää yhteenvedon projektista, jotta voit tarkistaa tekemäsi valinnat. AppWizardilla ei voi enää muuttua luotua projektia. Tässä vaiheessa riittää, kun luet yhteenvedon ja painat **OK**. Projekti luodaan ja se avautuu uuteen työtilaan.
10. **MyApp Classes** -kuvake ilmestyy Workspace-ikkunaan. Napauttamalla kuvakkeen plus-merkkiä pääset tutkimaan AppWizardin luomia luokkia. Seuraavassa luvussa nämä luokat käännetään toimivaksi sovellukseksi. Kaksoisnapauttamalla **CMyApp Class** -kuvaketta näet lähdekoodin luokkamäärittelyt.

## Oppitunnin yhteenveto

Lähdekoodi ja resurssitiedostot, jotka muodostavat sovelluksen tai komponentin ryhmitellään *projektiksi*. Projektit helpottavat tiedostojen käsittelyä ja sovelluksen rakentamiseen liittyvien asetusten hallintaa. Voit määritellä haluamasi määrän itsenäisiä *asetuskonfiguraatioita* projektillesi. Toisiinsa yhteydessä olevat projektit voidaan ryhmitellä samaan *työtilaan*.

Visual C++ tarjoaa joukon projektivelhoja, joiden avulla projektille tehdään runko, jonka pohjalta voidaan kehittää erityyppisiä sovelluksia ja komponentteja eri tekniikoita käyttäen.

MFC AppWizardin avulla voit luoda nopeasti ja helposti joukon luokkia ja resursseja, jotka voidaan kääntää toimivaksi ja ajettavaksi sovellukseksi. Käyttämällä AppWizardia voit:

- Luoda SDI-, MDI- ja dialogipohjaisia sovelluksia.
- Lisätä tietokanta-, ActiveX- tai OLE-tuen sovellukseesi.
- Määritellä sovelluksen ominaisuuksia, kuten työkalupalkkeja tai tilannekohtaisia ohjeita.
- Valita yhden useista erikoistuneista kantaluokista sovelluksesi näkymäksi.

## Oppitunti 2: Ohjelmointiympäristö

Microsoft Visual Studio 6.0 tarjoaa ohjelmoijalle monipuolisen, visuaalisen ympäristön, jota nimitetään integroiduksi kehitysympäristöksi (*integrated development environment*, IDE). Yhteisen IDE:n jakavat Visual C++, Microsoft Visual J++ sekä Microsoft InterDev ja lisäksi se voi isännöidä näiden työkalujen tuottamia projekteja samoin kuin muitakin tiedostotyyppejä.

Tällä oppitunnilla tutustut joihinkin Visual C++:n kehitysympäristön ominaisuuksiin, joita tyypillisesti käytetään sovelluksia kehitettäessä.

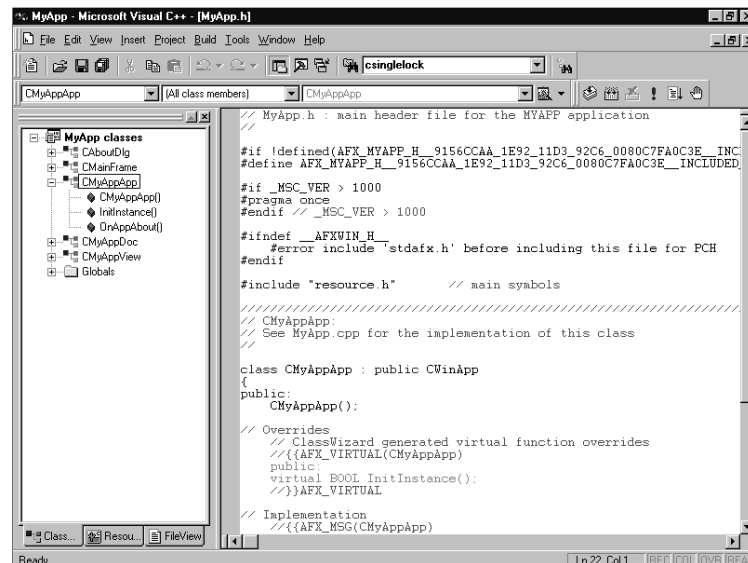
### Tämän oppitunnin jälkeen:

- Tunnet Visual C++:n käyttöliittymän elementit mukaan lukien sen ikkunat, valikot ja työkalurivit ja sen, kuinka niitä voi konfiguroida ja mukauttaa.
- Tiedät, kuinka saat ohjeita ohjelmointiympäristössä.
- Tiedät, kuinka muutetaan projektin asetuksia.
- Osaat kääntää MyApp-projektin.

**Oppitunnin arvioitu kesto: 30 minuuttia**

## Visual C++:n käyttöliittymä

Kuvassa 2.11 näet MyApp-projektin työtilan sellaisena kuin se oli oppitunnin 1 lopussa. Se on jaettu kahteen pääosaan: työtilaikkunaan ja muokkausikkunaan.



**Kuva 2.11** Visual C++:n käyttöliittymä

*Työtilaikkuna* esittää graafisesti työtilassa olevat objektit käyttäen hierarkkista puukontrollia. Kuten näit oppitunnilla 1, työtilassa olevat projektit muodostavat tämän puurakenteen juuret. Työtilaikkuna tarjoaa kolme erilaista näkymää projektiisi. Näkymää voit vaihtaa käyttämällä ikkunan alareunassa olevia välilehtiä:

- **ClassView** Esittää objektorientoituneen näkymän hyödyntäen C++ luokkia, niiden jäsenmuuttujia ja jäsenfunktioita. Minkä tahansa tällaisen kohteen kaksoisnapauttaminen tuo näkyviin sen määrittelyn tai toteutuksen lähdekoodissa.
- **ResourceView** Esittää resurssiobjektit ryhmiteltynä luokittain. Minkä tahansa kohteen kaksoisnapauttaminen avaa tarvittavan editorin.
- **FileView** Näyttää kaikki projektisi muokattavissa olevat tiedostot.

Kohteen napauttaminen oikealla painikkeella avaa pikavalikon, jossa on joukko kohteeseen liittyviä toimintoja.

*Muokkausikkuna* on alue, jossa työskentelet avoimien tiedostojen kanssa, joko ohjelmaeditorilla tai jollain resurssieditorilla. Jos haluat lisätilaa muokkaamisen aikana, napauta **Full Screen** -komentoa **View**-valikossa.

## Valikot ja työkalurivit

Visual C++ -ohjelmointiympäristö tarjoaa käytettäväksesi koko joukon valikoita, joiden avulla voit hallita työtilassa olevia tiedostoja ja projekteja, muuttaa ympäristön asetuksia, sekä avata ohjejärjestelmän, lähdekoodin hallinnan ja muita ulkoisia työkaluja. Useilla valikoilla on niitä vastaava työkalupalkki, josta voit käynnistää toiminnon yhdellä napautuksella. Visual C++:n työkalupalkit ovat täysin muokattavissa. Voit määrittää, mitkä työkalupalkeista ja mitkä niiden painikkeista ovat näkyvissä. Tämä tarkoittaa sitä, että voit mukauttaa ympäristöä niin, että useimmin käyttämäsi työkalut ovat nopeimmin saatavilla.

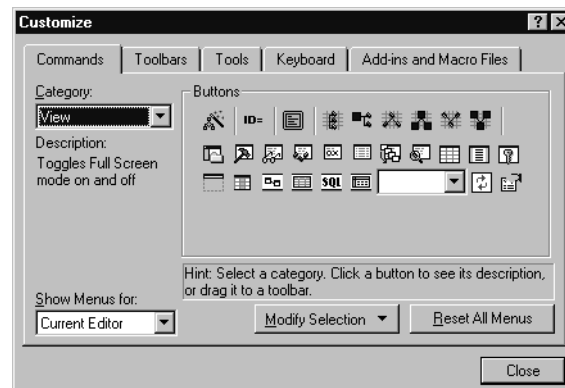
Uuden asennuksen jälkeen Visual C++:ssa on näkyvillä kolme tavallisimmin käytettyä työkalupalkkia. **Standard**-palkki sisältää tavallisimmat tiedostojen käsittelyyn liittyvät komennot; **Build**-minipalkki sisältää useimmin käytetyt ohjelman kääntämiseen ja ajamiseen liittyvät toiminnot ja **WizardBar** tarjoaa käyttöön luokkien kanssa työskenneltäessä tarvittavat komennot.

Hiiren kakkospainikkeen napauttaminen minkä tahansa työkalupalkin tyhjän alueen päällä tuo esiin luettelon käytettävissä olevista työkalupalkeista. Voit valita listasta, mitkä palkeista ovat näkyvissä ja mitkä eivät.

Seuraava harjoitus havainnollistaa sitä, kuinka voit mukauttaa käyttöliittymää ja kertoo, kuinka voit vaihtaa kokoruudun muokkaustilaan ja takaisin normaaliin yhdellä hiiren napautuksella tai näppäimen painalluksella.

► **Painikkeen lisääminen työkaluriville**

1. Valitse **Tools**-valikosta **Customize**. **Customize**-dialogi avautuu.
2. Napauta **Commands**-välilehteä.
3. Valitse **Category**-luettelosta **View**.
4. Napauta **Full Screen** -kuvaketta (kolmas vasemmalta ylimmällä kuvakerivillä) ja huomaa, että komennon kuvaus ilmestyy **Category**-luettelon alapuolelle. Dialogin pitäisi näyttää nyt samanlaiselta kuin kuvassa 2.12.



**Kuva 2.12** Customize-dialogin **Commands**-välilehti

5. Raahaa **Full Screen** -kuvake Standard-työkaluriville.
6. Sulje **Customize**-dialogi.
7. Avaa lähdekooditiedosto kaksoisnapauttamalla tiedoston nimeä FileViewssä.
8. Kokeile uutta painiketta napauttamalla sitä, jolloin näet lähdekoodin koko näytöllä. Kun haluat siirtyä takaisin tavalliseen tilaan, napauta painiketta uudelleen.

---

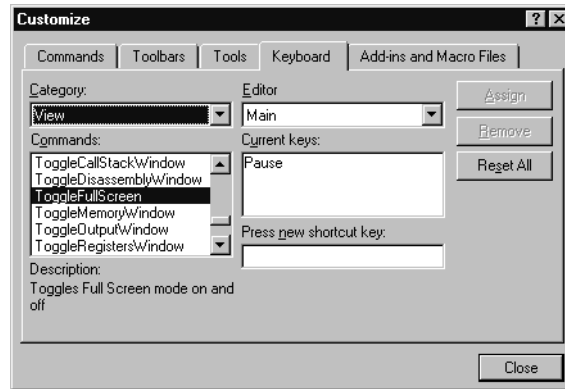
**Huomio** **Customize**-dialogin ollessa auki voit poistaa painikkeita työkalupalkista napauttamalla niitä hiiren oikealla painikkeella ja valitsemalla **Delete**-komennon avautuvasta pikavalikosta. Voit poistaa painikkeen myös raahaamalla sen ulos työkalupalkista ja pudottamalla sen mihin tahansa muualle kuin toisen työkalupalkin päälle.

---

► **Pikanäppäimen käyttöönotto**

1. Valitse **Tools**-valikosta **Customize**. **Customize**-dialogi avautuu.
2. Napauta **Keyboard**-välilehteä.
3. Valitse **Category**-luettelosta **View**.
4. Valitse **Commands**-ruudusta **ToggleFullScreen**. Huomaa, että komennon kuvaus ilmestyy **Category**-valikon alle.

5. Napauta **Press new shortcut key** -muokkausruutua ja paina näppäimistöltä PAUSE.
6. Napauta **Assign**. Dialogin tulisi olla nyt kuvan 2.13 mukainen.

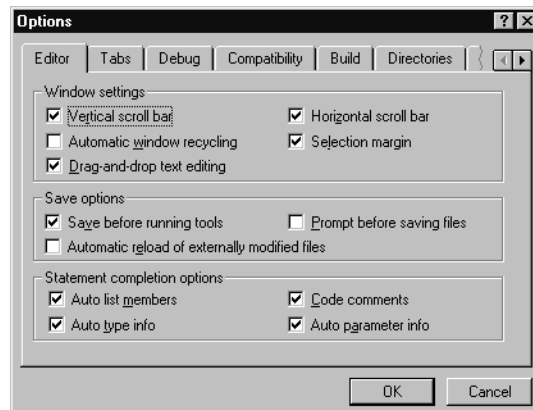


Kuva 2.13 Pikinäppäimen käyttöönotto

7. Sulje **Customize**-dialogi.
8. Kokeile mitä tapahtuu, kun painat PAUSE-näppäintä. Muokkausikkunan pitäisi vaihdella **full screen** -tilasta normaaliin ikkunaan ja takaisin.

## Ympäristön asetusten muuttaminen

Avaa **Options**-dialogi (kuvassa 2.14) napauttamalla **Options**-komentoa **Tools**-valikossa. Tämä välilehdillä jaettu ikkuna mahdollistaa Visual C++ -työskentely-ympäristön asetusten muuttamisen. Nämä valinnat muuttavat työskentely-ympäristön muodostavien integroitujen työkalujen ulkonäköä ja käyttäytymistä vaikuttaen kaikkiin projekteihin ja projektiasetusten konfiguraatioihin.



Kuva 2.14 Options-dialogi

**Options**-dialogin avulla voidaan konfiguroida:

- Koodieditorin toimintoja perustoiminnoista kuten ikkunan asetuksista ja tallennustoiminnoista erikoistointeihin kuten automaattiseen koodin täydennykseen ja syntaksiväriytykseen. Editori voidaan jopa mukauttaa emuloimaan muita suosittuja editoreja kuten BRIEF:ä.
- Debuggerissa näkyvää tietoa.
- Lähdekoodinhallinnan asetuksia.
- Hakemistopolkuja, joista tiedostoja haetaan ladattaessa suoritettavia ohjelmia ja etsittäessä lähdekooditiedostoja, includetiedostoja ja kirjastotiedostoja.

## Ohjeen käyttö

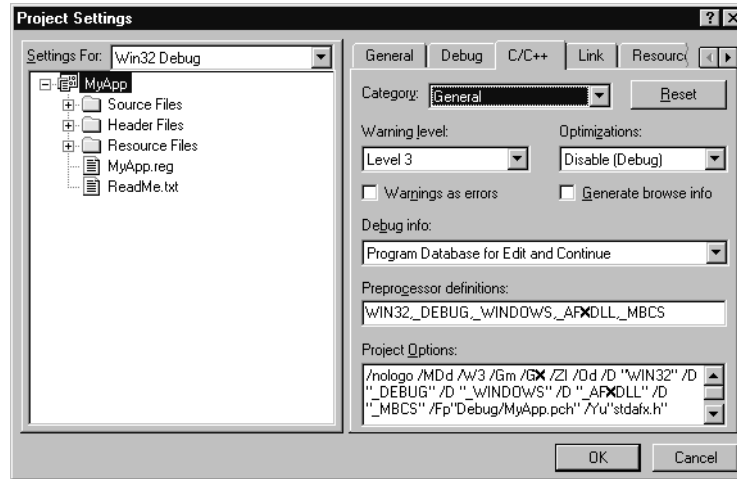
Visual Studio 6.0:n mukana toimitetaan sen oma versio Microsoft Developer Network (MSDN) -kirjastosta. Se on kehittäjille välttämätön tietolähde, joka sisältää yli gigatavun teknistä ohjelmointitietoa, kuten dokumentaatiota, teknisiä artikkeleita, mallikoodia, Microsoft Knowledge Basen ja kaikkea muuta sellaista, mitä saatat tarvita kehittäessäsi sovelluksia Microsoftin teknologialla.

MSDN-kirjasto toimii omassa HTML-pohjaisessa ympäristössään, joka on täydennetty nopealla ja älykkäällä hakukoneella. Se on integroitu C++-ohjelmointiympäristöön niin, että tilanteesta riippuva ohje on aina saatavilla tekstieditorista (ja tulosteikkunoista) painamalla F1-näppäintä.

Microsoft on kehittänyt lisäavuksi Automatic Statement Completion -toiminnon, joka on tilanteesta riippuva automaattinen koodintäydennystoiminto. Tämä ominaisuus, joka tunnetaan myös nimellä IntelliSense, tuo MFC-, Win32- ja ATL-kirjastot käsiisi esittämällä luokkien jäseniä, funktioiden prototyyppejä, määrittelyjä ja koodin kommentteja kohdassa, jossa olet kirjoittamassa koodia. IntelliSense näyttää välittömästi käytettävissä olevat mahdollisuudet. IntelliSense voi myös täydentää tunnistetut sanat, helpottaen näin pitkien luokkien tai niiden jäsenten nimien kirjoittamista.

## Projektiasetusten määrittely

Avaa **Project Settings** -dialogi, joka on kuvassa 2.15 seuraavalla sivulla, napauttamalla kohtaa **Settings Project**-valikossa. Tämän dialogin avulla voidaan muuttaa monia määrittelyksiä, mukaan lukien Visual C++:n kääntäjän ja linkittäjän asetukset.



**Kuva 2.15** Project Settings -dialogi

Millä tahansa välilehdellä mihin tahansa asetukseen tehty muutos vaikuttaa vain **Settings For** -listassa valittuna olevaan projektiin ja sen käännettyihin versioihin. Jokaisen projektin jokaisella käännettävällä osalla on omat asetuksensa. Jos haluat käyttää samoja asetuksia projektin kaikille osille, valitse **All configurations** -vaihtoehto **Settings For** -listasta. Kun muutat projektin asetuksia, tarkista huolella, minkä osan asetuksia muutat. On helppoa muuttaa väärän osan asetuksia, jos et ole varovainen.

#### ► Projektin asetusten muuttaminen

1. Valitse **Project**-valikosta **Settings**. **Project Settings** -dialogi avautuu.
2. Napauta **C/C++**-välilehteä.
3. Valitse **Category**-luettelosta **C++ Language**.
4. Valitse **Settings For** -ruudusta **All Configurations**. Huomaa, että vain ne asetukset, jotka ovat kaikille konfiguraatioille yhteisiä, näkyvät ominaisuussivun **Common Options** -ruudussa.
5. Saat RTTI:n käyttöön kaikissa MyApp-projektin konfiguraatioissa valitsemalla **Enable Run-Time Type Information (RTTI)**. Huomaa, että /GR -valitsin lisätään **Common Options** -ruutuun.
6. Tallenna muutokset napauttamalla **OK**-painiketta **Project Settings** -dialogissa.

Kun asetuksia muutetaan projektitasolla, pitää vasemman ikkunan puunäkymästä valita oikea projekti. Voit käydä läpi projektin luokittelukansiot ja yksittäisiä tiedostoja luokittelukansioissa. Dialogin ominaisuuslomakkeella näkyvät asetukset vastaavat valittuna olevan objektin asetuksia. Yleinen asetus tiedosto- tai kansiotasolla on tietyn tiedoston tai tiedostojen sulkeminen pois käännöksestä.



**Debug**, **C/C++** ja **Link**-välilehdet sisältävät oikeastaan usean sivun asetukset yhdellä lehdellä. Voit siirtyä asetusryhmästä toiseen valitsemalla yhden kyseisen välilehden **Category**-luettelon vaihtoehtoista.

Asetuksia, joita voit **Project Settings** -dialogia käyttäen muuttaa, ovat:

- **General Settings** Täällä voit määrittää, linkitetäänkö MFC:n staattiset kirjastot mukaan suoritettavaan ohjelmaan (jos muutat mieltäsi sen jälkeen, kun olet ohittanut vaiheen 5 AppWizardissa). Voit myös määrittellä kansiot luotavia tiedostoja varten.
- **Debug Settings** Täältä voit antaa komentoriviargumentteja ohjelmallesi, jos se käynnistetään debuggerissa. Voit myös käyttää uudelleenohjauskomentoja kuten komentokehotteessa.
- **C/C++ Settings** Täällä voit määrittellä kääntäjän yleiset asetukset, kielen ominaisuudet, kutsutoiminnot, prosessoriin liittyvät asetukset, optimoinnit, esikääntäjän määrittelyt ja niin edelleen.
- **Linker Settings** Täällä voit määrittellä lisäkirjastoja linkitettäväksi suoritettaviin ohjelmiisi.

## Projektin kääntäminen

Nyt, kun tunnet työskentely-ympäristön paremmin, voit kääntää MyApp-sovelluksen, jonka teit oppitunnilla 1.

### ► MyApp.exe:n kääntäminen

1. Valitse **Build**-valikosta **Set Active Configuration**. Varmista, että **MyApp—Win32 Debug** on valittu ja napauta **OK**.

---

**Vihje** Kun työskentelet useiden projektien ja konfiguraatioiden kanssa, kannattaa käyttää kokonaista **Build**-työkaluriviä **Build**-minirivin sijasta. **Build**-työkalurivissä on valikko, josta voit nopeasti ja vaivattomasti vaihtaa projektista tai konfiguraatiosta toiseen. Se myös muistuttaa sinua siitä, mitä projektia tai konfiguraatiota olet käsittelemässä.

---

2. Valitse **Build**-valikosta **Build MyApp.exe**  
tai  
Napauta **Build**-minityökaluriviltä (kuvassa 2.16) **Build**  
tai  
Paina F7.



**Kuva 2.16** Build-minityökalurivin **Build**-painike

Projekti käännetään ja linkitetään. MyApp.exe-tiedosto luodaan Debug-kansioon.

► **Ohjelman kokeileminen**

1. Valitse **Build**-valikosta **Execute MyApp.exe**

tai

Napauta **Build**-minirivin **Execute Program** -painiketta (punainen huutomerkki, joka on toinen painike oikealle **Build**-painikkeesta)

tai

Paina CTRL+F5.

Huomaa, että *tulosteikkuna* (output window) ilmestyy ruudun alareunaan. Tulosteikkunassa on useita näkymiä, joiden välillä voit liikkua alareunassa olevia välilehtiä käyttämällä. Sopiva näkymä tuodaan yleensä automaattisesti esille riippuen siitä, minkälaisia tulosteita ympäristö tuottaa. Kun käännösprosessi etenee, kääntäjältä ja linkittäjältä tulevat viestit näkyvät **Build**-välilehdellä.

2. Kun käännös on saatu valmiiksi, testataan AppWizardin automaattisesti luomia toimintoja. Valitsemalla **Open**-komenon **File**-valikosta näet kuinka tiedostodialogi avautuu. Sinun täytyy toteuttaa tiedoston avaamis- ja tallennusrutiinit itse.
3. Valitse **Print Preview** -toiminto **File**-valikosta ja näet, että esikatselu-arkkitehtuuri on jo valmiina. Tiedot, jotka tekemäsi koodi tuo näykyviin sovelluksesi pääikkunassa, nähdään esikatselussa siinä muodossa kuin ne näkyvät paperille tulostettuina.
4. Sulje Print Preview -ikkuna napauttamalla **Close**, ja sulje MyApp-sovellus valitsemalla komento **Exit File** -valikosta.

## Oppitunnin yhteenveto

Visual Studio 6.0 tarjoaa ohjelmoijille monipuolisen, visuaalisen, integroidun kehitysympäristön (IDE, integrated development environment).

Visual C++:n uusi asennus järjestää projektin valmiiksi. Valikkojen ja työkalupalkkien ohella käyttöliittymä jaetaan kolmeen osaan: työtilaikkunaan, muokkausikkunaan ja tulostusikkunaan. Työtilaikkuna tarjoaa erilaisia näkymiä projektisi sisältöön. ClassView esittää objektorientoituneen näkemyksen C++-luokkien, sekä niiden jäsenmuuttujien ja funktioiden avulla. ResourceView näyttää resurssiobjektit luokittain ryhmiteltynä. FileView näyttää kaikki muokattavissa olevat tiedostot projektissasi.

Visual C++ sisältää suuren joukon työkalupalkkeja, joita voit piilottaa, tuoda näkyville ja muokata tarpeidesi mukaan. Voit muokata työkalupalkkeja käyttäen **Tools**-valikon **Customize**-komentoa.

Visual C++ -ympäristö on täysin muokattavissa. Voit muokata ympäristön asetuksia käyttämällä **Tools**-valikon **Options**-komentoa.

Visual Studio 6.0:n mukana toimitetaan MSDN-kirjasto, joka on arvokas tietolähde kaikille ohjelmoijille.

Visual C++:ssa projektin asetukset voidaan muuttaa keskitetysti yhdestä paikasta. Voit muuttaa Visual C++:n käyttämän kääntäjän ja linkittäjän asetuksia **Project Settings** -dialogia käyttäen. Voit määritellä haluamasi määrän erilaisia konfiguraatioita projektin asetuksille.

## Oppitunti 3: Lähdekoodin hallinta

Lähdekoodin hallinta mahdollistaa lähdekooditiedostoihin tehtyjen muutosten seuraamisen ja on tärkeä työkalu työryhmässä ohjelmoitaessa. Visual C++ antaa mahdollisuuden lähdekoodin hallintajärjestelmän integroimiseen ohjelmointiympäristöön.

Tällä oppitunnilla opit, kuinka lähdekoodin hallintajärjestelmä, tässä tapauksessa Visual Studio 6.0:n Microsoft Visual SourceSafe (VSS) 6.0, otetaan käyttöön ja kuinka Visual SourceSafen ominaisuuksia käytetään Visual C++ -ohjelmointiympäristöstä.

---

### Tämän oppitunnin jälkeen:

- Tiedät, kuinka Visual SourceSafea käytetään lähdekoodin hallintaan.
- Tiedät, kuinka Visual SourceSafe Administratoria käytetään uuden tietokannan luomiseen, uuden käyttäjän lisäämiseen ja salasanojen ylläpitoon.
- Tiedät, kuinka Visual C++ -ohjelmointiprojektin lähdekoodi otetaan hallintaan.
- Tiedät, kuinka Visual C++:n lähdekoodin hallintatoiminnot yksinkertaistavat ohjelmoijan lähdekoodin ylläpitoon liittyviä tehtäviä.

**Oppitunnin arvioitu kesto: 30 minuuttia**

---

## Visual SourceSafen käyttöönotto

Ylläpitääkseen lähdekoodin hallintajärjestelmältä vaadittavan turvallisuustason ja yhdenmukaisuuden VSS tallentaa kopion lähdekoodistasi tietokantaan. Kaiken tyyppiset tiedostot, mukaan lukien tekstitiedostot, grafiikkatiedostot, binaaritiedostot, äänitiedostot ja videotiedostot, voidaan varastoida järjestelmään. Kun lisäät tiedoston VSS:ään, se tallennetaan tietokantaan ja asetetaan muiden saataville. Ryhmäsi jäsenet voivat nähdä minkä tahansa tiedoston viimeisimmän version, tehdä muutoksia ja tallentaa uuden version tietokantaan. Tiedostoon tehdyt muutokset tallennetaan niin, että vanha versio voidaan palauttaa koska vain.

Lähdekoodinhallintajärjestelmän käyttäjänä sinulla on *vain-luku*-kopio lähdekooditiedostosta työhakemistossa paikallisella kiintolevyllä. Kun haluat tehdä muutoksia tiedostoon, *kuittaat ulos* (check out) tiedoston, joka on peräisin VSS tietokannasta. Työhakemistossa oleva tiedosta korvautuu muokattavalla versiolla ja muita käyttäjiä estetään korvaamasta nykyistä versiota uudella. Kun olet tehnyt kaikki tarvittavat muutokset, kuittaat tiedoston *palautetuksi* (check in), jolloin muokaamasi tiedoston uusi versio tulee muiden ohjelmoijien käytettäväksi. Jos haluat perua tekemäsi muutokset, voit perua kuittauksen, ja

näin korvata paikallisesti muuttamasi tiedoston tietokannassa säilytetyllä versiolla.

Kun haluat varmistaa, että sinulla on viimeisimmät versiot kaikista projektin tiedostoista, käytä **Get Latest Versions** -komentoa kopioidaksesi projektin lähdekooditiedostojen nykyiset versiot työhakemistoosi.

Perusasetusten mukaan käyttäjä ei voi kuitata ulos tiedostoa, jonka joku toinen on jo kuitannut. Tällä varmistetaan se, että vain yksi henkilö kerrallaan tekee muutoksia tiedostoon. Oletetaan, että ohjelmoijilla A ja B on mahdollisuus muokata samaa tiedostoa samanaikaisesti. Jos A tallentaa tekemänsä muutokset ennen kuin B tallentaa omansa, kaikki A:n tekemät muutokset katoavat. VSS:n ylläpitäjä voi muuttaa asetuksia niin, että samanaikaiset kuitaukset ovat mahdollisia ja kuitenkin toisten tekemien muutosten ylikirjoittaminen on estetty. Tämä vaatii ylläpitäjältä hieman lisätyötä, sillä tämän täytyy käyttää VSS:n **Merge**-toimintoa varmistaa, että samanaikaiset muutokset eivät ole ristiriidassa keskenään.

Kun kuittaa jonkin tiedoston muokattavaksesi, sinun tulee varmistaa, että sinulla on käytössäsi viimeisin versio kaikista tiedostoista. Saatat muuten tehdä muutoksia, jotka tekevät koodistasi yhteen opimattoman muiden äskettäin tehtyjen muutosten kanssa. Voit käyttää VSS:n **Get Latest Version** -komentoa tiedostojen uusimpien versioiden hakemiseen työhakemistoosi.

Tiedostot on järjestetty VSS-tietokannassa projekteiksi. Projektit sisältävät tavallisesti kaikki ohjelmistokomponentin tekemiseen tarvittavat osat, mutta projekti voidaan määrittää myös millä tahansa muulla käyttäjän kannalta järkevällä tavalla. Projekteilla on taipumus olla järjestettyinä samaan tapaan kuin tiedostokansioilla.

Tavallisesti VSS-tietokannalle määrätään ylläpitäjä, joka on siitä vastuussa. Ylläpitäjä käyttää Visual SourceSafe Administrator -ohjelmaa määritellään tietokannan sijainnin, ylläpitäessään käyttäjäluetteloa ja käyttöoikeuksia, sekä varmuuskopioidessaan tietokannan.

Visual SourceSafe Explorer on apuohjelma, joka toimii käyttäjän liittymänä VSS-projekteihin. Monet yleisimmistä tehtävistä — tiedostojen kuittaaminen, uusimpien versioiden noutaminen jne — voidaan kuitenkin suorittaa suoraan Visual C++ -ympäristöstä. VSS:n integrointi Visual C++:aan itse asiassa yksinkertaistaa lähdekoodin hallintaprosessia ohjelmoijan kannalta.

---

**Huomio** Visual C++:n lähdekoodin hallintaominaisuudet on suunniteltu toimimaan Visual SourceSafen lisäksi myös muiden tuotteiden kanssa. Mitä tahansa lähdekoodin hallintajärjestelmää, joka noudattaa Microsoftin standardia lähdekoodin hallintaliittymää, voidaan hyödyntää suoraan Visual C++:n valikoista.

---

## VSS tietokannan käyttöönotto

Seuraavassa harjoituksessa käytät VSS Administrator -ohjelmaa ottaaksesi VSS-tietokannan käyttöön oppitunnilla 1 luomaasi MyApp-koodia varten. Jos et jo ole asentanut VSS Workstation -komponentteja, sinun tulisi tehdä se nyt.

### ► VSS Workstation komponenttien asentaminen

1. Käynnistä Setup.exe-ohjelma Visual Studio CD:n 1 juurihakemistosta.
2. Valitse **Workstation Tools And Components**. Napauta **Next**.
3. Napauta Visual Studio setup -näytössä **Add/Remove**.
4. Valitse Visual Studio maintenance -näytössä **Microsoft Visual SourceSafe 6.0**. Asenna VSS napauttamalla **Continue**.
5. Kun asennus on valmis, sulje Visual Studio setup -ohjelma.

VSS Workstation -asennus asentaa yksityisen VSS tietokannan ja VSS asiakas-ohjelman tietokoneellesi. Todellisessa kehitysryhmäympäristössä asennettaisiin VSS Server -komponentit ja luotaisiin jaettu VSS-tietokanta turvalliselle, säännöllisesti varmuuskopioitavalle palvelimelle.

### ► MyVSS-tietokannan luominen

1. Avaa **Käynnistä**-valikko, avaa **Ohjelmat**, avaa **Microsoft Visual Studio 6.0**, avaa **Microsoft Visual SourceSafe** ja napauta **Visual SourceSafe 6.0 Admin**.
2. Kirjaudu VSS:een tunnuksella **Admin** ilman salasana. Jos saat varoituksen siitä, että Admin-tilillä ei ole salasanaa, ohita ilmoitus napauttamalla **OK**.
3. Valitse **Tools**-valikosta **Create Database**. **Create New VSS Database** -dialogi avautuu.
4. Kirjoita muokkausruutuun **C:\Source Control\MyVSS**. Jätä valintaruutu valituksi. MyVSS-tietokanta ja kaikki sen tarvitsemat hakemistot luodaan määrättyyn paikkaan, kun napautat **OK**.

### ► MyVSS-tietokannan avaaminen

1. Valitse **Users**-valikosta **Open SourceSafe Database**.
2. Napauta **Open SourceSafe Database** -dialogista **Browse**. Etsi ja avaa **File**-dialogin avulla srcsafe.ini-tiedosto kansiota C:\Source Control\MyVSS.
3. Napauta **OK**. MyVSS-tietokannan tulisi ilmestyä **Open SourceSafe Database** -dialogiin. Varmista, että se on valitu ja napauta **Open**.
4. Napauta **OK**.
5. Huomaa, että tietokannan nimi ilmestyy ikkunan otsikkopalkkiin.

- **Admin-tilin salasanan asettaminen**
  1. Varmista, että **Admin**-käyttäjä on valittu.
  2. Valitse **Users**-valikosta **Change Password**.
  3. Jätä **Change Password** -dialogin **Old Password** -ruutu tyhjäksi. Kirjoita ja vahvista uusi salasana.
- **Käyttäjätunnuksen lisääminen**
  1. Valitse **Users**-valikosta **Add User**.
  2. Kirjoita **Add User** -dialogissa käyttäjätunnus ja salasana. Huomaa, että voit valintaruudulla määritellä, että käyttäjällä on vain luku -oikeus. Jätä se valitsematta.
  3. Napauta **OK**.

## MyApp-projektin ottaminen lähdekoodin hallintaan

Seuraavaksi lisätään oppitunnilla 1 tehty projekti **MyVSS**-tietokantaan. Toimenpide on helpointa tehdä Visual C++ -ohjelmointiympäristöstä.

- **MyApp-projektin lisääminen MyVSS-tietokantaan**
  1. Käynnistä Visual C++. Avaa MyApp-työtila.
  2. Napauta hiiren kakkospainikkeella pääikkunan tyhjää aluetta. Valitse **Source Control**.
  3. **Source Control** -työkalurivi, kuvassa 2.17, avautuu. Seuraavissa harjoituksissa viitataan tämän työkalurivin painikkeisiin. Voit vaihtoehtoisesti käyttää vastaavia **Project**-valikon **Source Control** -alivalikon komentoja.



Kuva 2.17 Source Control -työkalurivi

4. Napauta **Add To Source Control** (kuten kuvassa 2.17).
5. **Visual SourceSafe Login** -ruutu avautuu. Varmista, että **MyVSS** näkyy **Database**-ruudussa. Kirjoita luomasi käyttäjänimi ja salasana.

6. Kun napautat **OK, Add To SourceSafe Project** -dialogi avautuu ja näyttää **MyApp**-projektin nimen. Ota MyApp-projekti lähdekoodin hallintaan napauttamalla **OK** (ja luo uusi projekti napauttamalla **Yes**).

---

**Huomio** \$/-symboli viittaa tietokannan *juuriprojektiin* (root project). Kaikki VSS-tietokannassa olevat projektit ovat juuriprojektin aliprojekteja.

---

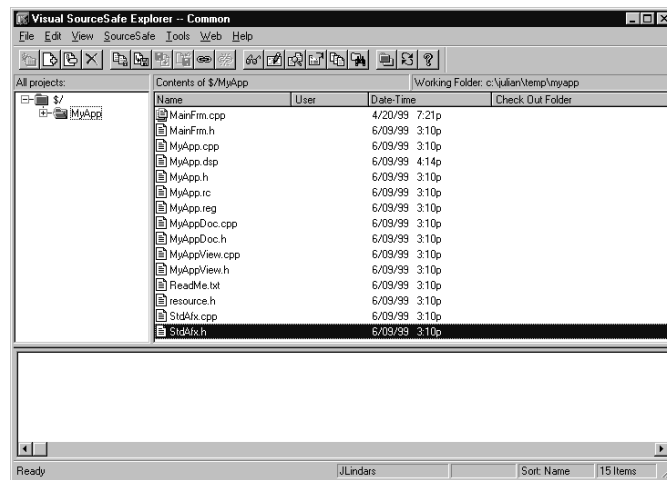
7. **Add To Source Control** -dialogi avautuu ja siinä ovat kaikki projektin tiedostot valittuina. Lisää valitut tiedot lähdekoodinhallintaan napauttamalla **OK**.

## VSS Explorer

Seuraavassa harjoituksessa tutkitaan uutta VSS-projektia käyttämällä VSS Exploreria.

### ► MyApp-projektin katseleminen

1. Käynnistä VSS Explorer (kuvassa 2.18) napauttamalla **Source Control** -painiketta, joka on **Source Control** -työkalurivin oikeassa reunassa.



**Kuva 2.18** Visual SourceSafe Explorer

2. Kokeile käyttööliittymää avaamalla kaikki kansiot ikkunan vasemmassa osassa ja katso mitä tiedostoja projektiin on sijoitettu. Tutki valikoita ja käytä työkaluvihjettä apuna tunnistaksesi työkalupalkin painikkeet.

### ► Tiedoston uloskuittaminen

1. Napauta vasemmassa osassa MyApp-kansiota.
2. Napauta oikeassa osassa ReadMe.txt-tiedostoa.



3. Valitse **SourceSafe**-valikosta **Check Out** (tai käytä työkalurivin vastaavaa painiketta).
4. Kuittaa ReadMe.txt-tiedosto napauttamalla **OK**. Huomaa, kuinka Visual SourceSafe Explorer näyttää kuittauksen tilan.
5. Siirry takaisin Visual C++:aan. Katso projektitiedostoja FileView-ikkunassa. Huomaa, että tiedostokuvakkeet ovat nyt harmaita sen merkiksi, että ne ovat lähdekoodin hallinnassa. Huomaa, että ReadMe.txt on merkitty osoituksena siitä, että se on kuitattu käytettäväksi.
6. Palaa VSS Exploreriin.

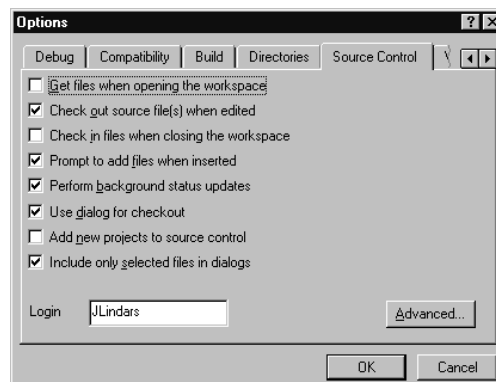
► **Tiedoston palauttaminen**

1. Napauta VSS Explorerin oikeassa osassa **ReadMe.txt**-tiedostoa.
2. Voit valikon ja työkalurivin painikkeen lisäksi kuitata ReadMe.txt-tiedoston palautetuksi napauttamalla sitä hiiren kakkospainikkeella ja valitsemalla pikavalikosta **Check In**.
3. Palauta ReadMe.txt-tiedosto napauttamalla **OK**.

Sulje VSS Explorer ja palaa Visual C++:aan. Haluat ehkä nyt kokeilla samojen toimintojen toistamista Visual C++ -kehitysympäristössä. Huomaat, että toimenpiteet ovat käytännössä samanlaiset. Voit käyttää **Project**-valikon **Source Control** -alivalikon toimintoja, **Source Control** -työkalupalkin painikkeita tai voit napauttaa hiiren oikealla painikkeella FileView-ikkunan kuvaketta.

## Lähdekoodin hallinnan asetukset

Visual C++:aan integroituja lähdekoodin hallintatoimintojen asetuksia voidaan muuttaa **Tools**-valikosta avattavan **Options**-ominaisuussivun **Source Control** -välilehdellä (kuvassa 2.19). Huomaa, että sinun täytyy mahdollisesti käyttää nuolipainikkeita saadaksesi **Source Control** -välilehden näkyviin.



Kuva 2.19 Source Control -välilehden asetukset

Kuvassa 2.19 nähdään lähdekoodin hallinnan alkuperäiset asetukset. Nämä asetukset vaikuttavat työskentelyyn Visual C++:lla seuraavasti:

- **Check out source files when edited** Jos yrität muokata tiedostoa, jota et ole kuitannut muokausta varten, **Check Out** -dialogi avautuu ja antaa mahdollisuuden suorittaa kuittauksen. (Huomaa, että sinun ei sallita muokata tiedostoja, joita et ole kuitannut.)
- **Prompt to add files when inserted** Kun lisäät tiedostoja projektiisi, sinua kehoitetaan lisäämään ne lähdekoodin hallintaan.
- **Perform background status updates** Lähdekoodin hallinnan seuraamien tiedostojen tilatiedot päivitetään taustalla eikä välittömästi muutoksen tapahduttua.
- **Use dialog for checkout** Kun kuitaat tiedoston palautetuksi suoraan ClassViewistä, ResourceViewistä tai FileViewistä, dialogi avautuu kehottaen sinua vahvistamaan toimenpiteen. Muussa tapauksessa kaikki tiedostot, joita toimenpide voisi koskea, listataan ikkunaan.
- **Include only selected files in dialogs** Kun päätät suorittaa toimenpiteen (kuten **Add to Source Control** tai **Check In**) valituille tiedostoille ClassViewissä, ResourceViewissä tai FileViewissä, vain valittuna olleet tiedostot ilmestyvät dialogiin. Muussa tapauksessa kaikki tiedostot, joita toimenpide voisi koskea, listataan.

Voit tehdä myös seuraavat asetukset:

- **Get files when opening the workspace** Kun tämä toiminto on asetettu, VSS kysyy, haluatko hakea niiden tiedostojen, joita et ole kuitannut muokattavaksesi, uudet versiot työtilaa avattaessa. Voi olla, ettet halua tehdä näin, varsinkin jos projektisi on edelleen työn alla.
- **Check in files when closing the workspace** Kun tämä toiminto on asetettu, VSS kehottaa sinua kuittaamaan palautetuiksi kaikki tiedostot, jotka olet ottanut muokattavaksi samalla kun suljet Visual C++:n työtilan. Tämä ei myöskään ole välttämättä toivottavaa, jos työskentelysi on kesken, sillä toiset käyttäjät saattaisivat saada keskeneräiset versiot tiedostoistasi.
- **Add new projects to source control** Kun luot uuden projektin tämän toiminnon ollessa käytössä, sinua kehoitetaan lisäämään uusi projekti lähdekoodin hallintaan.

Jos napautat **Advanced**-painiketta tässä ikkunassa, saat näkyviin asiaankuuluvan joukon lähdekoodin hallintaohjelman lisäasetuksia.

## Visual SourceSafe-tiedostojen uudelleen käyttö

Voit uudelleen käyttää tiedostoja sekä samassa VSS-projektissa että toisissa VSS- projekteissa. Tiedostoja uudelleen käytettäessä tarvittavia komentoja ovat esimerkiksi **Share**, **Branch** ja **Merge**.

### Jakaminen

VSS:ssä yksi tiedosto voidaan jakaa (share) usean projektin kesken. Tiedostoon yhdessä projektissa tehty muutos näkyy automaattisesti myös muissa samaa tiedostoa käyttävissä projekteissa. Esimerkiksi oletetaan, että haluat jakaa kaikki MyApp-projektin header-tiedostot muiden projektien kanssa. Jakaaksesi tiedoston luot jakolinkin kahden (tai useamman) projektin välille. Sama tiedosto ilmestyy näin molempiin projekteihin samanaikaisesti. Kun kuittaat tiedoston palautetuksi mistä tahansa projekteista, tekemäsi muutokset päivitetään automaattisesti kaikkiin muihin linkitettyihin tiedostoihin.

#### ► Kaikkien MyApp-projektin header-tiedostojen jakaminen toisten projektien kanssa

1. Avaa Visual SourceSafe Explorer.
2. Luo toinen projekti, jonka kanssa jaat tiedostoja, napauttamalla hiiren oikealla painikkeella **MyApp**-projektin kuvaketta ja napauta **Create Project** -painiketta.
3. Kirjoita **Create Project** -dialogissa projektin nimeksi **NewProject** ja napauta **OK**.
4. Napauta juuri luotua **NewProject**-kuvaketta.
5. Valitse **SourceSafe**-valikosta **Share**.
6. Valitse **MyApp**-projekti **Share with** -dialogin projektiosasta.
7. Kirjoita **File to share** -muokkausruutuun \*.h ja paina ENTER, jolloin saat näkyviin kaikki MyAppin header-tiedostot.
8. Valitse *kaikki* luettelossa olevat header tiedostot. (Valitse listan ensimmäinen tiedosto, paina ja pidä pohjassa SHIFT näppäintä samalla, kun valitset listan viimeisen tiedoston nimen.) Napauta **Share**.
9. Napauta **Close**. Huomaa, että jaetut tiedostot on lisätty NewProject-projektiin ja niillä on erityinen kuvake ilmaisemassa, että ne ovat jaettuja tiedostoja.

#### ► Kaikkien saman tiedoston jakavien projektien esille tuominen

1. Napauta hiiren kakkospainikkeella yhden jaetun tiedoston kuvaketta. Valitse pikavalikosta **Properties**.
2. Napauta ominaisuussivun **Links**-välilehteä. Projektit, jotka jakavat tämän tiedoston, ilmestyvät luetteloruutuun.

## Haarottaminen

Normaalisti, kun jaat tiedoston, siitä on olemassa vain yksi versio. Saatat kuitenkin haluta luoda mukautetun kopion tiedostosta jotain tiettyä projektia varten. Oletetaan esimerkiksi, että haluat luoda mukautetun version MyAppDoc.h-tiedostosta NewProjectia varten. Tehdäksesi tämän sinun täytyy käyttää VSS:n *haarotus* (branching) -ominaisuutta. Haarottaminen luo tiedostosta kaksoiskappaleen yhden projektin tiedostosta ja käyttää kaksoiskappaletta luodessaan uuden version projektista. Näitä kaksoiskappaleita voidaan muuttaa vaikuttamatta alkuperäisiin tiedostoihin. Myöhemmässä vaiheessa voit käyttää yhdistämistoimintoa (merge) tiedostoihin tehtyjen muutosten yhdistämiseen ja jälleen luoda yhdistetyn version tiedostosta.

► **Mukautetun version tekeminen MyAppDoc.h-tiedostosta NewProject-projektiin**

1. Valitse Visual SourceSafe Explorerissa **NewProject**-projekti. Valitse oikealta **MyAppDoc.h**-tiedosto napauttamalla.
2. Valitse **SourceSafe**-valikosta **Branch**.
3. Napauta **OK**. Huomaa kuinka kuvake muuttuu osoituksena siitä, että tiedosto ei ole jaettu.

## Yhdistäminen

Kuten edellä mainittiin, tiedostot, jotka on haarotettu rinnakkaisiksi versioiksi, voidaan yhdistää (merge) takaisin yhdeksi tiedostoksi. Oletetaan esimerkiksi, että kehitysryhmä on tehnyt muutoksia alkuperäiseen MyApp-projektin MyAppDoc.h-tiedostoon ja sinä haluaisit liittää tehdyt muutokset myös NewProject-projektissa olevaan mukautettuun versioon. Yhdistämistoiminto vertaa eroja kahden tiedoston välillä ja luo yhden tiedoston, jossa kaikki tehdyt muutokset ovat mukana.

► **MyAppDoc.h-tiedoston erillisten versioiden yhdistäminen**

1. Valitse Visual SourceSafe Explorerissa **NewProject**-projekti. Valitse oikealta **MyAppDoc.h**-tiedosto.
2. Valitse **SourceSafe**-valikosta **Merge Branches**.
3. Valitse MyApp-projekti **Merge to \$/NewProject/MyAppDoc.h**-dialogissa. Huomaa, että yhdistettävät versiot näytetään **Versions**-sarakeessa. Napauta **Merge** ja vapauta **OK** avautuvassa sanomaikkunassa.

Yhdistäminen tulee yleisimmin tarpeelliseksi, jos ylläpitäjä sallii useamman käyttäjän kuitata saman tiedoston muokattavaksi samanaikaisesti, mahdollistaen näin saman lähdekooditiedoston rinnakkaisen muuttamisen. Käyttäjä voi lisäksi yhdistää paikallisen kopion ja uusimman VSS-tietokantaan tallennetun version käyttämällä **Get Latest Version** -komentoa.

## Versiohistorian seuraaminen

VSS:ssa on useita toimintoja, joiden avulla voit seurata versiohistoriaa. **Show History** -toimintoa käytetään yleisesti, kun halutaan yhteenveto tiedoston versiohistoriasta.

### ► MyAppDoc.h-tiedoston versiohistorian tutkiminen

1. Valitse **NewProject**-projekti Visual SourceSafe Explorerissa. Napauta hiiren kakkospainikkeella ikkunan oikeanpuoleisesta osasta **MyAppDoc.h** -tiedostoa.
2. Valitse pikavalikosta **Show History**.
3. Hyväksy oletukset napauttamalla **OK**. Tiedoston versiohistoria avautuu.
4. Sulje History-ikkuna.

## VSS-projektin arkistointi

Visual SourceSafe Administrator -ohjelman avulla SourceSafe-projektin arkistointi on helppoa. Valitsemalla **Archive** valikon **Archive Projects** -toiminnon, voit varmuuskopioida projektitiedostosi pakattuun tiedostoon. Voit käyttää **Archive Projects** -toimintoa **Restore Projects** -komennon kanssa siirtäessäsi tiedostoja VSS-tietokannasta toiseen.

### ► MyApp-projektin arkistointi

1. Avaa VSS Administrator -ohjelma. Aikaisemmin luodun **MyVSS**-tietokannan pitäisi avautua automaattisesti.
2. Valitse **Archive**-valikosta **Archive Projects**.
3. Valitse **Choose Project to Archive** -dialogista **MyApp**-projekti ja napauta **OK**.
4. Napauta **Next**, kun MyApp-projekti on näkyvissä **Archive Wizardin Projects to Archive** -ruudussa.
5. Valitse Archive Wizardin vaiheessa 2 **Save data to file** -toiminto ja aseta arkistotiedoston nimeksi **C:\SourceControl\backup.ssa**. Napauta **Next**.
6. Valitse vaiheessa 3 **Archive all of the data** ja napauta **Finish**.
7. Kun arkistointi on valmis, sulje VSS Administrator -ohjelma.

Kun haluat palauttaa tiedot arkistosta, avaa ensin tietokanta, jonka haluat palauttaa **VSS Administrator** -ohjelmalla. Valitse sitten **Restore Projects** -toiminto **Archive**-valikosta ja käytä **Restore Wizard** -velhoa arkiston sisältävän .ssa-tiedoston etsimiseen ja lataamiseen.

## Oppitunnin yhteenveto

Lähdekoodin hallinta mahdollistaa lähdekooditiedostoihin tehtyjen muutosten seuraamisen ja on tärkeä osa ryhmässä tapahtuvaa kehitystyötä. Visual SourceSafe (VSS) 6.0, joka toimitetaan Visual Studio 6.0 mukana, on lähdekoodin hallintajärjestelmä, joka on täysin integroitu Visual C++ -ohjelmointiympäristöön.

VSS tallentaa keskitetysti kopiot lähdekooditiedostoista tietokantaan. Kehitysryhmän jäsenet voivat nähdä viimeisimmän version mistä tahansa tiedostosta, tehdä muutoksia ja tallentaa uuden version tietokantaan. Tieto tehdyistä muutoksista tallennetaan, joten vanha versio on palautettavissa koska vain.

Lähdekoodin hallintajärjestelmän käyttäjillä on *vain-luku*-kopio lähdekooditiedostoista paikallisen kiintolevyn työhakemistossa. Kun haluat muokata tiedostoa, voit Visual SourceSafe Explorer -ohjelman avulla kuitata tiedoston ulos VSS-tietokannasta. Toimenpide hakee muokattavan kopion tiedostosta työhakemis-toosi. Kun halutut muutokset on tehty, kuittaa tiedoston palautetuksi ja muut ohjelmoijat saavat käyttöönsä uuden, vasta muokatun version tiedostosta. Varmistaaksesi, että sinulla on uusimmat versiot kaikista projektin tiedostoista, käytä **Get Latest Versions** -komentoa, joka kopioi nykyiset versiot projektin lähde-kooditiedostoista työhakemistoosi.

VSS Administrator Program -ohjelma mahdollistaa kehitysryhmäsi projektien luomisen ja hallitsemisen. Voit tämän ohjelman avulla kontrolloida pääsyä tietokantaan ja arkistoida projekteja.

Visual SourceSafe on täysin integroitu Visual C++ -kehitysympäristöön. Visual C++:n voidaan säätää tekemään mitä tahansa lähdekoodinhallintaan liittyviä tehtäviä automaattisesti — voit esimerkiksi säätää järjestelmän niin, että aina kun li-säät tiedostoja projektiin, saat kehoitteen niiden lisäämisestä lähdekoodin hallintaan.

Visual SourceSafen **Share**-komento mahdollistaa tiedoston yhden kopion jakamisen useiden projektien kesken. VSS:n **Branch**-komento mahdollistaa saman tiedoston eri versioiden rinnakkaisen muokkaamisen ja **Merge**-komennon avulla voidaan kahteen eri versioon tehtyt muutokset yhdistää takaisin yhdeksi tiedostoksi.

**Show History** on yksi monista tavoista, joilla voit seurata tiedoston versiohistoriaa.

## Laboratorio 2: STUupload-projektin luominen

Tässä harjoituksessa käytetään MFC AppWizardia luvussa 1 esitellyn STUupload.exe-sovelluksen kehitysprojektin luomiseen. Voit käyttää tätä projektia pohjana seuraavien lukujen laboratorioharjoituksissa.

### ► STUupload-projektin luominen

1. Avaa Visual C++.
2. Valitse **File**-valikosta **New**.
3. Valitse **MFC AppWizard (exe)** -vaihtoehto.
4. Kirjoita projektin nimeksi **STUupload**.
5. Napauta **OK**.
6. STUupload sallii käyttäjän käsitellä vain yhtä dokumenttia kerrallaan. Vaihda asetus **Multiple Documents** -vaihtoehdosta **Single Document** -vaihtoehtoon. Siirry vaiheeseen 2, tietokantatuki, napauttamalla **Next**.
7. STUupload mahdollistaa kyselyjen tekemisen tietokantaan, mutta toiminnon toteutuksessa käytetään ActiveX Data -objekteja. Tästä syystä MFC:n tietokantatukea ei tarvita. Jätä vaihtoehto **None** valituksi ja siirry vaiheeseen 3, yhdistettyjen dokumenttien toiminnot, napauttamalla **Next**.
8. STUupload-sovellukseen ei sisällytetä tukea yhdistelmäasiakirjoille, joten anna **None**-vaihtoehdon olla valittuna ja siirry vaiheeseen 4, sovelluksen ominaisuudet, napauttamalla **Next**.
9. Hyväksy tämän dialogin oletusarvot ja siirry **Advanced Options** -dialogiin napauttamalla **Advanced**-painiketta.
10. Kirjoita **Document Template Strings** -sivun **File Extention** -ruutuun tiedoston tarkentimeksi **stu**. Sulje **Advanced Options** -dialogi napauttamalla **Close**.
11. Muuta **Filter Name**, **File New Name** ja **File Type Name** -ruutuihin sanan **STUplo** tilalle sana **STUupload**.
12. Sulje **Advanced Options** -dialogi napauttamalla **Close** ja siirry vaiheeseen 5 napauttamalla **Next**.
13. Jätä oletusarvot voimaan ja siirry AppWizardin vaiheeseen 6 napauttamalla **Next**.
14. STUupload-sovellukseen tehdään vieritettävä näkymä, jotta katseluikkuna voisi olla näytön alaa suurempi. Mahdollistaaksesi tämän, periytä näkymäluokkasi MFC:n **CScrollView**-luokasta, joka sisältää vierityspalkkien ja vieritystoiminnon toteutuksen. Varmista, että **CSTUuploadView** on valittuna ruudun yläosassa ja napauta **CScrollView**-vaihtoehtoa **Base Class** -alasvetovalikosta.

15. Napauta **Finish**, tarkista **New Project Information** -ruutu ja luo projekti napauttamalla **OK**.
16. Kun projekti on luotu, laajenna STUload Classes -kuvake ClassView -ikkunassa ja tarkastele luokkia, jotka AppWizard on sinulle luonut. Käännä projekti painamalla F7.
17. Käynnistä STUload-sovellus painamalla CTRL+F5. Valitse **File**-valikosta **Open**. Huomaa, että tiedostonavausdialogi odottaa sinun avaavan .stu-päätteisiä STUload-tiedostoja, kuten AppWizardin vaiheessa 4 määriteltiin. Sulje **Open File** -dialogi napauttamalla **Close**.
18. Siirrä hiirtä käyttämällä STUload-sovelluksen ikkunan oikeaa alareunaa, niin että ikkuna kutistuu muutaman neliösentin kokoiseksi. Huomaat, kuinka vierityspalkit ilmestyvät automaattisesti ikkunan reunoille.
19. Sulje STUload-sovellus valitsemalla **File**-valikosta **Exit**.



## Kertaus

1. Mitä projektin asetuksia MFC AppWizard tekee automaattisesti?
2. Missä vaiheessa voit säätää sovellusikkunan ominaisuuksia AppWizardissa?
3. Millaisia ohjeita on saatavilla Visual C++:n IDE:ssä?
4. Mitä asetuksia voit tehdä **Project Settings** -dialogin C/C++-välilehdellä?
5. Mistä syystä saatat joutua käyttämään **Visual SourceSafen Get Latest Version** -komentoa?
6. Missä tilanteissa tarvitaan Visual SourceSafen **Merge**-toimintoa?