

Avoimien tiedostojen kopiointi

FSBackup-ohjelmalla avoimetkin tiedostot talteen

Avoimien tiedostojen kopiointi on usein hankalaa ellei mahdollista. Miksi ihmeessä ei levyopinasta saada repäistyä tiedostoa kuin tiedostoa talteen, mikäli se on käytössä? Ongelmaan löytyy ratkaisu tällä pienellä mallitiedostolla, joka kopioi tiedoston paikasta toiseen, vaikka tiedosto olisi käytössä.

Varmistusohjelmien käyttäjät ovat törmänneet aikojen alusta samaan ongelmaan avoimista tiedostoista ja niiden varmistamisesta. Varmistusohjelma ei varmista avoimia tiedostoja, vaikka tiedostoon ei kirjoiteta koskaan mitään ohjelman käynnin aikana. Tällaisia ovat mm. asetuksia sisältävät tiedostot. Varmistuksen kyydistä tippuu näin oleellisia tiedostoja, joiden sisältönä on vaikka asetuksia tai muita ohjelman toimintaan vaikuttavia toimintoja.

Järjestelmäkutsut

Itse taistelin USA:sta kotoisin olevan ohjelmiston kanssa, jonka SQL-tietokanta oli 24h/vrk auki, mutta käytössä ainoastaan aamu-kuudesta iltakymmenen. Ohjel-

man olisi voinut varmistaa öisin, mutta ketäpä olisi tällöin ajamassa ohjelmaa alas ja käynnistämässä varmistusta. Kallis SQL-kantojen käytönaikaisen varmistuksen suorittava ohjelma oli liian arvokas kukkarolle.

Niinpä uppouduin hetkeksi MS-Windowsin ja Visual Basicin manuaaleihin ja ohjelmointiin. Pienen tutkinnan jälkeen löytyi sopiva käyttöjärjestelmän kutsu, jota hyödyntämällä saisi tiedoston kuin tiedoston kopioitua. Järjestelmäkutsut eivät olleetkaan enää salaperäisiä kesäjuhlina järjestelmien osajille!

Ohjelman kasausta

Ohjelman kasaukseen käytin Visual Basic -kääntäjää. Delphin omistajille ei tuottane vaikeuksia muuntaa ohjelmaa sillekin. Ohjelmassa oli aluksi vain pari ikkunaa lähde- ja kohdetiedoston nimille, mutta hetken päästä käytin Mitä- ja Minne-listoja, joihin ohjelma luki tiedot asetustiedosta BCList.txt. Nimetkää tarvittaessa omanne mielenne mukaan.

Seuraavaksi mukaan liitettiin GetFileList-toiminto, jolla ohjelma osasi hakea kokonaisen hakemiston tiedostonimet – kaikkien tiedostonimien kirjoittaminen BCList.txt:in kyllästytti laiskaa miestä, joten *- ja ?-valitsimia toteuttava GetFileList oli tarpeen.



Ohjelman ruudussa on vain kolme tekstikenttää: suoritettu, seuraava ja aikaväli. Painonappi "Suorita" käynnistää kopiointin. Mitä- ja minne-listoihin täytetään Copy-käskyä varten lähde- ja kohdetiedostot. Timer eli ajastin tutkii viiden sekunnin välein, onko seuraava suoritus aika on saavutettu.

Suoritettu-kentässä on viimeinen suoritushetki ja Aikaväli-kentän verran kun lisätään tunteja, saadaan Seuraava-kenttään suorituksen ajankohta.

Timer-ajastin käyttääkin, olisiko jo suorituksen paikka eli onko kello ohittanut suorituksen ajankohdan. Tällöin ajastin painaa Suorita-nappia eli kutsuu Suorita_Clickiä.

Lopuksi lisäsin vielä vanhalla perinteisellä tavalla kirjoituksen BCLini.txt:in, johon talletetaan Suoritettu-, Seuraava- ja Aikaväli-kentän arvot omille riveilleen. Ohjelma luo tarvittaessa ko. tiedoston, kuten myös päivittää tiedoston aina kun jostakin kentästä poistetaan.

Ohjelma muuten luo molemmat BCList.txt- ja BCLini.txt -tiedostot, ellei niitä löydy asennuspolun hakemistosta.

Käännöstä

Lopuksi vielä pari koeajoa ja varoittavia Msg_boxeja pomp-paamaan ruudulle, mikäli tiedostonimet ovat puuta heinää tai asetustiedostot puuttuvat, ja ohjelma oli valmis.

Kääntäjä rouhaisi hieman ja levyllä syntyi peräti 0,032 MB:n kokoinen eli 32 kB:n ohjelma.

Loppuviilauksena muutin oletuskooksi "Minimized" eli ohjelma näkyy tehtäväpalkissa ollessaan päällä! Ohjelma vaatii tuekseen MS Visual Basic Runtime 6.0:n, mutta itse ohjelma on pieni.

Mikroviestin purkissa lieneekin jo sorsat ja toimiva ohjelma ja asennuspaketti itse ohjelmalle.

Ohjelma ja VB6-runtime riittävät toimintaan, joten kaikkien ei tarvitse käänellä ja ohjelmoida koodia!

Heureka!

Ohjelma toimi toiveiden mukaan ja ylikin. SQLBase-tietokanta meni lokitiedostoineen varmistukseen, vaikka tietokantapalvelin oli päällä. Varmistukset jopa toimivat moitteitta.

BCList.txt:n käyttö käy näppärästi Notepadilla/Muistiolla, joka käynnistyy heti kun tiedostoa kaksoisklikataan.

Ohjelmaa voi kukin parannela tarpeen mukaan eli lisällä vaikka pakkausruutiinit tuottamaan ZIP-tiedostoja, BCList.txt:n täytön voi tehdä hiirellä toimivaksi ja kullekin tiedostolle tai hakemistolle voi luoda omat väli- ja varmistusväli- jne. Ohjelman voisi ohjelmoida käyttämään vaihdettavaa mediaa, esim. korppua, jolloin ohjelma paloittelisi isot tiedostot monelle levyille. Vaihtoehtoja olisi monia!

Lopuksi

Koska kyseessä on vapaasti käytettävää koodia, ei kirjoittaja ota vastuuta toimivuudesta tai ohjelman väärinkäytöstä. Esim. SQL-kantojen varmistus ohjelmalla ei onnistune aina, sillä ohjelma ei osaa estää kirjoitusta kopiointin aikana ts. lokikirjat voivat mennä sekaisin, jos tietokanta päättää kirjoittaa kopiointin aikana tms. Ohjelma on kuitenkin toimiva esimerkki moneen käyttöön.

Varsinkin yhdessä Zip-aseman kanssa on sillä helppo varmistaa halutut kriittiset hakemistot ja tiedostot Zip-levylle, vaikka seitsemällä levyllä päivittäin levyä vaihtamalla!

Näppärä koodinväntäjä saanee ohjelman toimimaan vaikka MS-Excelissä ja MSAccessissa tai muissakin Basic-tulkin omaavissa Microsoftin tuotteissa. Varmistava tekstinkäsittelyohjelma voisi olla aika veikeä toteutus!

Listaus 2. (Listaus 1 ladattavissa osoitteessa www.mikroviesti.net/apicopy)

Moduli ApiCopy.bas sisältää itse API-kutsun, jolla tiedosto kuin tiedosto repäistään levyllä riippumatta senaukiolosta.

Option Explicit

```
Declare Function apiCopyFile Lib "kernel32" Alias "CopyFileA" _
    (ByVal lpExistingFileName As String, _
    ByVal lpNewFileName As String, _
    ByVal bFailIfExists As Long) As Long
```

```
Sub CopyFile(SourceFile As String, DestFile As String)
    Dim Result As Long
    If Dir(SourceFile) = "" Then
        MsgBox Chr(34) & SourceFile & Chr(34) & _
            " ei ole kelvollinen tiedostonimi."
    Else
        Result = apiCopyFile(SourceFile, DestFile, False)
    End If
End Sub
```