

Snabbguide Visma Compact API

© Copyright 2006-2010 Visma Spcs AB

Visma Compact

2008

1 2 9 8 1 5 7
5 2 3 3 2 4 8
6 1 2 4 5 5 2 7
1 0 4 9 2 1 2 3 5 1 2
8 2 1 5 6 3

3 4 5 2 3 4 1 2 3 4 5 6 7 8 9 0
8 9 1 2 3 7 3 6 5 4 3 2 1 0
2 5 5 1 7 8 2 1 0 9 8 7 6 5 4 3 2 1 0
1 2 4 3 7 8 1 2 3 4 5 6 7 8 9 0
2 4 4 5 5 6 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4



VISMA[®]

9 2 1
5 6 8 2 1 5 6 3
8 2 3 2 6 5 4

Komma igång med Compact API

Hur APlet fungerar

Visma Compacts API läser och skriver direkt till databasen via ett antal färdiga metoder som innehåller mer eller mindre affärslogik. Detta innebär att du i din applikation på ett enkelt sätt kan skapa fakturor, reskontra, verifikat eller registerposter i Visma Compact utan att använda inläsningsfiler etc. Du får alltså en direktkoppling in i Visma Compact och APIet arbetar såsom en virtuell användare i programmet (kräver dock inte en egen användarlicens).

I stort sett alla metoder returnerar returvärden med felkod så det är lätt att fånga upp om fel uppstår.

Förberedelse

Börja med att installera Visma Compact. Vid installationen rekommenderar vi att du använder de sökvägar som föreslås. När Visma Compact är installerat måste du aktivera det. Det gör du i licensguiden som dyker upp när du startar programmet.

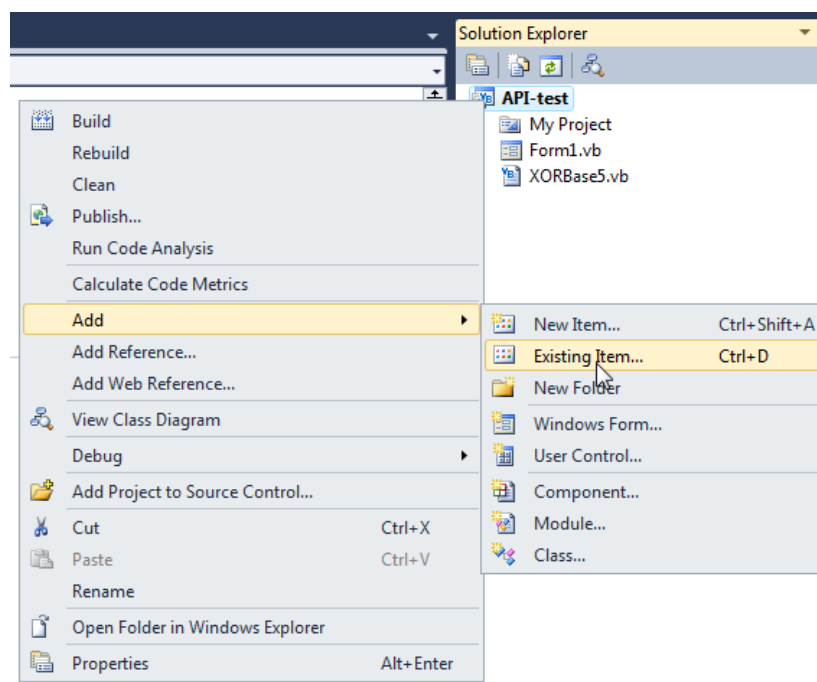
Om du utvecklar för eget bruk får du en licensnyckel till den version av Visma Compact som ditt avtal gäller.

Om du utvecklar för kommersiellt syfte erhåller du en licensnyckel som du ska använda.

Att lägga upp APlet i Microsoft Visual Studio 2010

Beskrivningen är i Microsoft Visual Studio 2010 och VB.NET som utvecklingspråk.

Skapa projektet och lägg till importfilen för VB.NET som heter Xorbase5.vb. Denna fil läggs till genom att du högerklickar på projektnamnet i Solution Explorer och väljer ”Add existing item...”. Därefter bläddrar du dig fram till filen och lägger till denna.

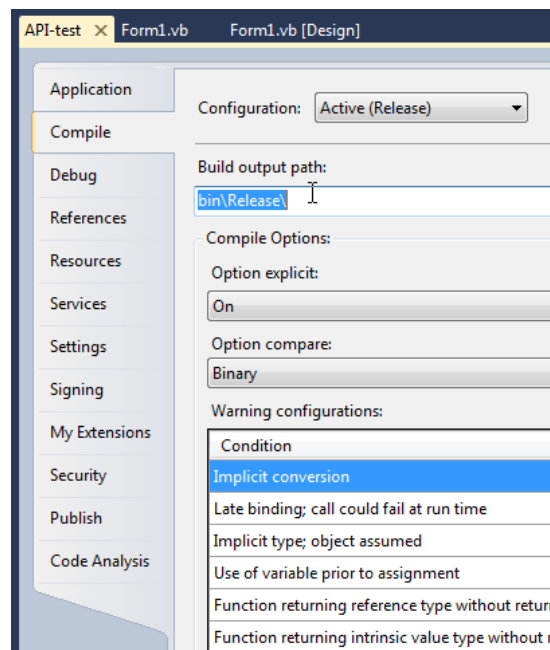


Visma Compact API Snabbguide

För att kunna exekvera din applikation måste följande filer göras tillgängliga för applikationen:

- C4DLL.DLL Databasmotorn
- XFILES.DLL Visma Compacts klassbibliotek
- XORBASE5.DLL Export av API-funktionerna

Detta kan göras genom att dessa kopieras till den sökväg som anges under Project / Compile / Build output path.



Högnivåanrop / Lågnivåanrop

APIet innehåller ett antal API-metoder med inbyggd affärslogik. Dessa högnivåmetoder är:

<i>InvoiceOpen</i>	<i>VoucherOpen</i>
<i>InvoiceAddRow</i>	<i>VoucherAddRow</i>
<i>InvoiceClose</i>	<i>VoucherClose</i>
<i>PersonAdd</i>	<i>LedgerAddCustomerInvoice</i>
<i>ArticleAdd</i>	<i>LedgerAddSupplierInvoice</i>
<i>ObjectAdd</i>	

Med dessa API-metoder behövs bara ett litet antal rader kod för att lägga upp poster i Visma Compact. Det är även enkelt att läsa poster från Visma Compact. Några exempel på metoder:

<i>InvoiceFind</i>	<i>InvoiceRowFind</i>
<i>VoucherFind</i>	<i>LedgerFind</i>
<i>PersonFind</i>	<i>ArticleFind</i>
<i>ObjectFind</i>	

Behöver man komplettera poster eller utföra operationer som inte stöds av ovanstående metoder så finns ett antal lågnivåmetoder att använda:

<i>RecordCreate</i>	<i>RecordFind</i>
<i>RecordFirst</i>	<i>RecordNext</i>
<i>RecordPrev</i>	<i>RecordLast</i>
<i>RecordUpdate</i>	<i>RecordAdd</i>
<i>RecordDelete</i>	

Se vidare i referensmanualen för mer information om vilka metoder som finns att använda och hur de används.

Ett enkelt kodexempel

Med Visma Compacts API-paket får du ett antal högnivåmetoder (med inbyggd affärslogik) som hjälper dig att skapa kundfaktura, reskontra, verifikat samt registerposter.

Det är till exempel inte svårare än så här att skapa en faktura:

```
Sub AddInvoice()  
  
    Dim Result As Integer  
  
    Result = DatabaseOpen("TESTBOLAG")  
  
    Result = InvoiceOpen("1001", XCUSTINVOICE, NOTPRINTED, 2010, 5, 30, 0, 0)  
  
    Result = InvoiceAddRow("1001", "", "", "", "", "st", 10, 1500, 0)  
    Result = InvoiceAddRow("1002", "", "", "", "", "st", 10, 1250, 0)  
    Result = InvoiceAddRow("1003", "", "", "", "", "st", 10, 3000, 0)  
  
    Result = InvoiceClose()  
  
    DatabaseClose()  
  
End Sub
```

(API-Metoderna är markerade i denna färg)

Lika enkelt skapar du en leverantörsreskontrapost, kundreskontrapost eller ett verifikat.

Utöver dessa metoder finns ett antal lågnivåmetoder för att läsa och skriva till databasen med mer eller mindre affärslogik inbyggt.

Ett mer utökat kodexempel

Du kan göra i stort sett allt i Visma Compact med APIet. Däremot rekommenderar vi inte att göra alltför avancerade operationer där det inte finns färdiga metoder för. Genom lågnivåanrop har du tillgång till hela databasen men ibland krävs det mycket god kunskap i affärslogiken för att det ska fungera. Tabellerna är nära knutna till varandra och Visma Compact förutsätter att dessa är rätt uppdaterade.

Nedan följer ett exempel på hur en leverantörsreskontrapost läggs in där man gör en kontroll om leverantören finns med i personregistret. Om denna inte finns i registret läggs den till. För att göra exemplet mer avancerat så läggs verifikatet in manuellt istället för att metoden lägger in ett standardverifikat. Även den inscannade fakturan läggs till så att denna kan läsas i dokumentspårningen i Visma Compact.

Visma Compact API Snabbguide

Principiellt skulle det räcka med följande enkla kod för att lägga till en leverantörsreskontrapost:

```
Sub LedgerAddInvoice()  
  
    Dim Result As Integer  
  
    Result = DatabaseOpen("TESTBOLAG")  
  
    Result = LedgerAddCustomerInvoice("1001", 2008, 1, 11, 294, 0, FA, 0, 1, "")  
  
    DatabaseClose()  
  
End Sub
```

Men liksom det enkla fakturaexemplet behöver man fånga upp returkoderna och eventuellt skriva kod för att t.ex. lägga till leverantören om denna saknas.

Ett mer utökat kodexempel:

```
Sub SupplierInvoice()  
  
    Dim OK As Boolean = True  
    Dim Result As Integer  
  
    Dim Amount As Double = 2500.0  
    Dim Vat As Double = 500.0  
    Dim Net As Double = Amount - Vat  
    Dim sID As String = "1009"  
    Dim sName As String = "Min leverantör"  
    Dim sAdr2 As String = "Stora gatan 10"  
    Dim sAdr3 As String = "123 45 STADEN"  
    Dim sTel As String = "040-123456"  
  
    Result = DatabaseOpen("TESTBOLAG")  
  
    If (Result = DbOk) Then  
  
        Dim kPerson1 As IntPtr = RecordKey(PERSON, 1)  
        Dim kPersType As IntPtr = KeyField(kPerson1, "TYPE")  
        Dim kPersNo As IntPtr = KeyField(kPerson1, "ID")  
  
        FieldAssignString(kPersType, "2")  
        FieldAssignString(kPersNo, sID)  
  
        If (RecordFind(kPerson1) = 0) Then 'Post saknas!  
            Result = PersonAdd(SUPPLY, sID, sName, "", "", sAdr2, sAdr3, "", sTel, "",  
                               "", -1, -1, -1)  
            If (Result < 0) Then  
                ' Felmeddelande med felkod i "Result"  
                OK = False  
            End If  
        End If  
    End If
```

Visma Compact API Snabbguide

```
If (OK) Then
  Result = VoucherOpen(3, 2010, 8, 21)
  If (Result > 0) Then
    Result = VoucherAddRow("2440", "FA/Linneas Maskinimport", "", "", -Amount)
    ' Lägga till eget verifikat
    ' Returvärdeskontroll bör ske här
    Result = VoucherAddRow("2640", "FA/Linneas Maskinimport", "", "", Vat)
    ' Returvärdeskontroll bör ske här
    Result = VoucherAddRow("5400", "FA/Linneas Maskinimport", "", "", Net)
    ' Returvärdeskontroll bör ske här

    ' Lägga till inskannad faktura
    DocumentAddSupplierInvoice("C:\Faktura.pdf", True)

    Result = LedgerAddSupplierInvoice(SID, "123", 2010, 8, 21, Amount, Vat,
                                      FA, 2, "")

    If (Result < 0) Then
      ' Felmeddelande med felkod i "Result"
    End If
  Else
    ' Felmeddelande med felkod i "Result"
  End If
End If
Else
  ' Felmeddelande med felkod i "Result"
End If
DatabaseClose()

End Sub
```

Visma Compact API FAQ

1. Vilka programmeringsspråk kan jag använda för att utveckla mot Visma Compact API?

Vi tillhandahåller exempelkod för språken VB.NET, Delphi, Visual Basic 6 samt C++. Utöver dessa går det att använda alla språk som kan referera till en C-dll.

2. Hur gör man för att utveckla i VB.NET miljö?

Importfilen för VB.Net heter Xorbase5.vb. Denna fil läggs enklast till efter att ett projekt skapas genom att man använder "Add existing item..." och sedan bläddrar sig till filen. För att kunna exekvera måste filerna C4DLL.DLL, XFILES.DLL och XORBASE5.DLL göras tillgängliga för applikationen. Detta kan t ex göras genom att de kopieras till den sökväg som anges i "Build output path" under projektets kompilerings inställningar.

3. Hur gör man för att utveckla i Delphi miljö?

Projekt utvecklade i Delphi måste använda sig av importfilen Xorbase5.pas. Denna läggs till med hjälp av kommandot "Add file to project...". Enhetsnamnet är xorbase5. För att kunna exekvera applikationen måste filerna C4DLL.DLL, XFILES.DLL och XORBASE5.DLL göras tillgängliga. Enklast görs detta genom att kopiera dem till samma mapp som projektet använder för applikationens exekveringsfiler.

4. Hur gör man för att utveckla i Visual Basic 6 miljö?

De projekt som fortfarande använder sig av Visual Basic 6 måste använda importfilen xorbase5.bas. Denna lägger man till sitt projekt t ex genom kommandot "Add module". För att kunna exekvera applikationen måste filerna C4DLL.DLL, XFILES.DLL och XORBASE5.DLL göras tillgängliga.

5. Hur gör man för att utveckla i C++ miljö?

API-programmering med C eller C++ är möjlig genom att inkludera importfilen xorbase5.h samt gör filerna filerna C4DLL.DLL, XFILES.DLL och XORBASE5.DLL tillgängliga. Filen Xorbase5.lib innehåller de funktionsdefinitioner som behövs för länkning.

6. Hur gör jag för att börja använda utvecklarpaketet mot Visma Compact?

Börja med att installera Visma Compact. Vid installationen rekommenderar vi att du använder de sökvägar som föreslås. När Visma Compact är installerat måste du aktivera det. Det gör du i licensguiden som dyker upp när du startar programmet.

Om du utvecklar för eget bruk får du en licensnyckel till den version av Visma Compact som ditt avtal gäller.

Om du utvecklar för kommersiellt syfte erhåller du en licensnyckel som du ska använda.

7. Hur fungerar felhanteringen?

I stort sett alla metoder returnerar returvärden med felkod så att du lätt kan fånga upp om fel uppstår.

8. Hur stänger man ner företaget och APIet korrekt?

Vid stängning av företaget används metoden DatabaseClose. Det är viktigt att denna metod anropas innan applikationen stängs. Resurser som allokerats för APIet frigörs då och databasen stängs ner korrekt.

9. Kan man ha flertrådade lösningar mot Visma Compact API?

APIet är inte utvecklat för att hantera flera samtida anrop. Det är inga problem om din applikation kör flera trådar, men det kan bara vara en tråd som arbetar mot APIet.

10. Är det tillåtet att läsa/skriva direkt till databasen via ODBC?

Både ja och nej. Vi tillåter inga ODBC kopplingar som skriver mot Visma Compacts databas. Däremot är det inga problem att använda ODBC för att läsa data från databasen för att t.ex. bygga upp en egen rapport eller dylikt. Vi har dock ingen support på ODBC koppling. Om man däremot skriver till databasen och fel uppstår i databasen tar vi inte ansvar för detta och kan neka supporthjälp alternativt ta extra betalt för denna support. Läs mer om detta i ert licensavtal.