

---

記事

[Hiroshi Sato](#) · 2020年7月2日 21m read

[Open Exchange](#)

## VisM.OCX(Caché Direct)を利用したアプリケーションをIRISに移行する方法

### 初めに

VisM.OCXはVisual Basicでクライアント・サーバー型のアプリケーション開発を支援するためにInterSystemsが提供してきたツールです。

誕生から既に20年以上が経過した非常に古いテクノロジーです。

OCX規格(ActiveXコンポーネント)は、マイクロソフト社が推進してきた規格ですが、やがてマイクロソフト社が後継となる.Net Frameworkをリリースし、その新しいフレームワークへの移行を強力に推進すると同時に、OCX規格は非推奨機能となっています。

一方で下位互換性のため、.Net Framework配下でOCXを動作可能とする仕組みが用意されており、結果としてOCXは、.Net Framework環境下で動作可能です。

従って、VisM.OCXも.Net対応のプログラミング言語C#やVB.NETからそのまま利用することができます。

しかしながら.Net Framework上で動作するとは言え、.Net Frameworkが用意する安全性の高い資源管理や強固なセキュリティ機能の恩恵を受けることができず、いわゆるマネージドコードとして動作できないという制約を抱えています。

上記の状況を整理した結果、InterSystems社は、IRISをリリースするに際し、VisM.OCXは、標準製品では動作せずに、特別なキーを適用することで動作可能になるような措置（使用を制限する）を適用しました。

その結果、VisM.OCXを使用するためには、InterSystems社にお問い合わせいただき、その特別なキーを取得するための申請を行う必要があります。

また広く皆様にIRISを使って評価していただきたいと考えて提供している無償のCommunityエディションでも動作しません。

ここでIRISでVisM.OCXアプリケーションを動かすのはハードルが高いと感じた方に朗報です。

VisM.OCXを使用せずに、VisM.OCX対応アプリケーションをIRIS上で動作可能にするための移行ツール（Caché Directエミュレーター）を用意しました。

このツールは、Open Exchangeから取得可能です。

[Open Exchangeサイト](#)

ここでは、このツールを使った移行作業を実際に簡単なVBアプリケーションを例として紹介していきたいと思います。

ここで説明しているADBKアプリケーションは、下記のGithubから入手可能です。

[ADBKサンプル](#)

## ADBKアプリケーション

このサンプルアプリケーションは、20年以上前にVB6サンプルとして作成されました。

アドレス帳をCaché データベースに保存し、クライアントのVB6アプリケーションからVisM.OCXを使って、データの登録や検索を行う非常にシンプルなサンプルです。

## VB6プロジェクトを.Netプロジェクトに変換する

このVB6のアプリケーションをまず.Netアプリケーションに変換しなければなりません。

この変換を行うためには、最新のVisual Studioではだめで、Visual Studio 2008を別途インストールする必要があります。


変換方法は以下のサイトをご参考ください。

[VB 6 アプリケーションをVB.NETに変換する方法](#)

## VisM.OCXの削除

次にその変換された.net プロジェクトファイルを新しいVisual Studio(現時点では2019が最新)で読み込みます。

まずメインフォームの右下に配置されているVisM.OCX ( 青い立方体のアイコン ) が見えると思います。



Form1

## 住所録

名

郵便番

住所:

自宅Tel

会社Tel

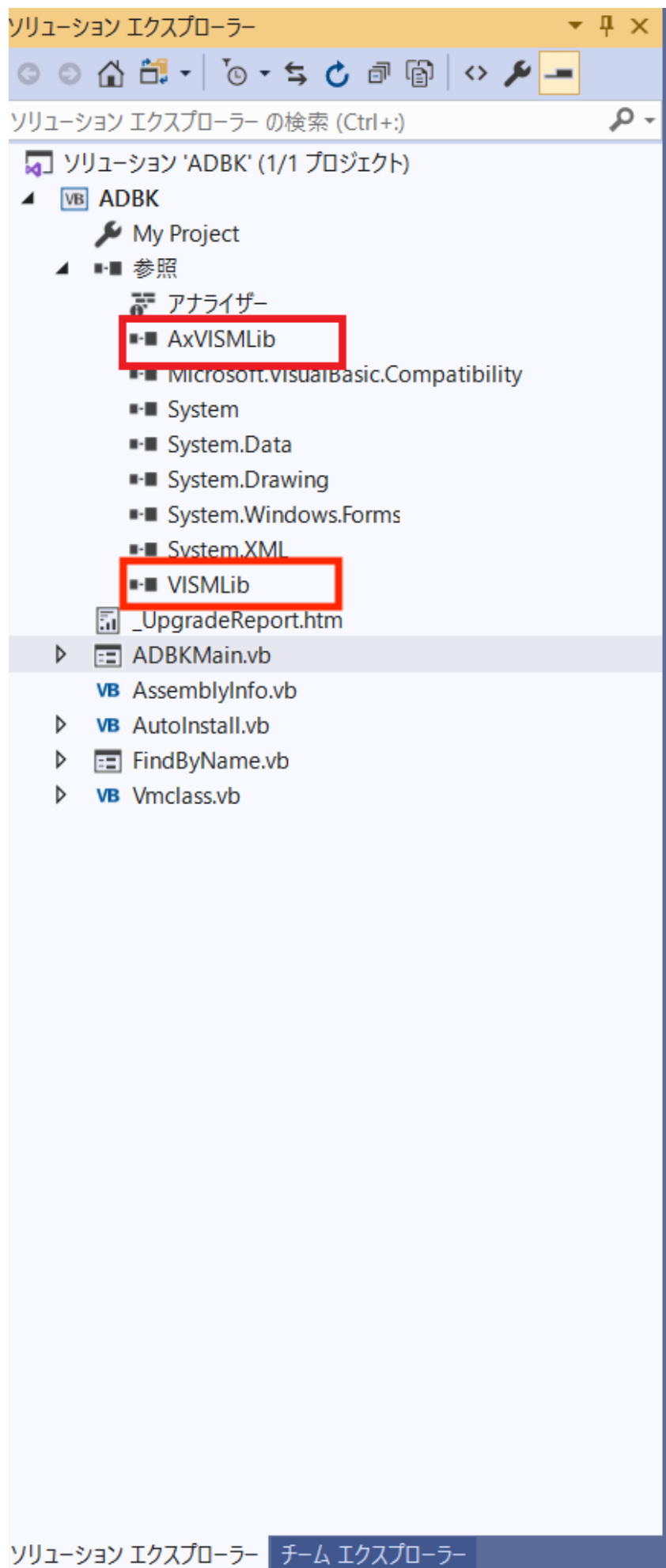
年齢:

生年月  YYYY-MM-

この部品を削除します。（選択した後に、右クリックで削除メニュー項目が表示される）

次に右側に表示されているソリューションエクスプローラーの参照の所を開いて、AxVISMLibを選択して右クリックし、削除を選びます。

同様にその下のVISMLibを選択して右クリックし、削除を選びます。



必ず部品の削除を先に行ってください。

ここで更新したファイルの保存を行い、プロジェクトを閉じます。

必要に応じて、この状態一式を別ディレクトリーにバックアップします。

## C#エミュレータクラスを使用可能にする

Caché DirectエミュレータをOpen

Exchangeからダウンロードし、そのZipファイルを適当なディレクトリーに展開します。

cacheDirectWrapper.csを新規のC# クラスライブラリープロジェクトに取り込みます。



参照設定でIRISの.Netライブラリーを指定します。

c::\InterSystems\IRIS\dev\dotnet\bin\4.6.2

InterSystems.Data.IRISClient.dll

プロジェクトメニューからxxxのプロパティをクリックします。

xxxはプロジェクト名です。  
左側のペインからアプリケーションを選びます。

アセンブリ名は適当な名前にします。( CacheDirectEmulatorなど )  
対象のフレームワークが.Net Framework 4.8でなければ、.Net Framework 4.8に変更します。

左側のペインからビルドを選びます。  
構成をアクティブな(Release)またはReleaseにします。

ビルドします。( dllが作成されます )

## Caché Directエミュレーターの参照設定を追加する

VB.Netのプロジェクトを再度開き、プロジェクトメニューから参照の追加を選択します。

作成されたdllを参照設定します。(dllとアプリケーションプロジェクトの.Net Frameworkのバージョンを合わせる必要があります)

## アプリケーション修正

それでは、VisM.OCX用に作られたアプリケーションをCaché Directエミュレータのインタフェースを使って動作するように変更しましょう。

まず、ADBKMain.vbを開きます。

まず、新しく追加したCaché Directエミュレータライブラリーを参照できるようにインポートする必要があります。

これは先頭にあるOption文の後ろに以下のように追加します。

```
Option Strict Off
Option Explicit On
'以下のインポートが必要
Imports cdapp
```

続いてVisM.OCXの置き換えです。

このVBアプリは、直接VisM.OCXを使用するのではなく、それをラップするクラスを作成していました。  
従って、ここでは、VisM.OCXを直接置き換えるのではなく、そのラッパークラスを置き換える方式を採用します。  
。

元々の定義は以下のようになっていました。

cVMClassというのが、VisMをラップしたクラスです。

```
Dim CacheDirect As New cVMClass
```

これをCaché Directエミュレータクラスを参照するように変更します。

長いので途中で折り返しています。

```
Public WithEvents CacheDirect As cacheDirectWrapper = New cacheDirectWrapper  
("Server = localhost; Port=1972; Namespace=USER; Password = SYS; User ID = system;")
```

VisM.OCXの場合は、サーバーへの接続は、いくつかオプションがあるのですが、このエミュレーターはオブジェクト生成時にサーバー接続まで行う仕様になっています。

WithEvents句は、イベント処理が不要の場合は、指定しなくてもいいです。

これでCacheDirectを参照しているところは、ほとんど修正することなくそのまま使用できます。

但しフォームのロード処理にいくつか変更が必要でした。

```
Private Sub ADBKMainLoad(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)  
Handles MyBase.Load  
    CacheDirect.VisM = VisM1.GetOcx  
    If Not Install("Address Book Demo Application", VisM1, "GLO", "^ADBK", "USER",  
"+$D(^$ROUTINE("ADBK"))", "adbk.gsa", "MAC", "ADBK", "USER", "+$D(^$ROUTINE("ADBK"))", "adbk.rsa")  
Then End  
End Sub
```

ここで元々のラッパークラスでは、VisM.OCXをプロパティとして設定する仕様だったのですが、これは今回必要なくなりました。

次にデータやルーチンが初期インストールされていない場合にそれらをロードする仕組みを用意していたのですが、これは今はちゃんとライブラリーが用意されていて、それを使うほうが簡単なので、書き換えることにします。

```
Private Sub ADBKMainLoad(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)  
Handles MyBase.Load  
    'CacheDirect.VisM = VisM1.GetOcx  
'If Not Install("Address Book Demo Application", VisM1, "GLO", "^ADBK", "USER", "+$D(^$ROUTINE("ADBK"))",  
"adbk.gsa", "MAC", "ADBK", "USER", "+$D(^$ROUTINE("ADBK"))", "adbk.rsa") Then End  
    CacheDirect.Execute("=$DATA(^$ROUTINE("ADBK"))")  
    If CacheDirect.VALUE = 0 Then  
  
        CacheDirect.P0 = "c:\temp"  
        CacheDirect.P1 = "ck-d"  
        CacheDirect.Execute("set P2=$system.OBJ.ImportDir(P0,,P1,.P2)")
```

```
If CacheDirect.P2.Substring(0, 1) <> 1 Then  
MsgBox("Loadでエラーが発生しました" & CacheDirect.P2)  
  
End If  
  
End If  
End Sub
```

最後にオブジェクトの消滅に関してもより安全な方法を採用したために、書き換えが必要です。

```
Private Sub ADBKMainFormClosed(ByVal eventSender As System.Object, ByVal eventArgs As  
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed  
'UPGRADENOTE: オブジェクト CacheDirect をガベージ  
コレクションするまでこのオブジェクトを破棄することはできません。 詳細については、'ms-  
help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6E35BFF6-CD74-4B09-9689-3E1A43DF8969"  
をクリックしてください。  
CacheDirect = Nothing  
End Sub
```

Caché Directエミュレータクラスのendメソッドを呼び出すように変更します。

```
Private Sub ADBKMainFormClosed(ByVal eventSender As System.Object, ByVal eventArgs As  
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed  
'UPGRADENOTE: オブジェクト CacheDirect をガベージ  
コレクションするまでこのオブジェクトを破棄することはできません。 詳細については、'ms-  
help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6E35BFF6-CD74-4B09-9689-3E1A43DF8969"  
をクリックしてください。  
'CacheDirect = Nothing  
CacheDirect.end()  
End Sub
```

続いてFindByName.vbの変更点を見てみましょう。

ここでもCacheDirectのインスタンス化を実施していましたが、今回はADBKMainクラスのCaché Directエミュレータクラスを共有できるので、ここで宣言する必要がなくなります。

以下の記述をコメントアウトします。

```
'Dim CacheDirect As New cVMClass
```

次に検索ボタンが押された時の処理を変更していきます。

```
Private Sub CmdFindClick(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)  
Handles CmdFind.Click  
  
Dim i As Object  
'Dim NOL As Short
```



```
' 名前のリストを取得
ListLookupName.Items.Clear()
'以下の機能は未実装
CacheDirect.Clear()
CacheDirect.PDELIM = Chr(1)
CacheDirect.P0 = "^ADBK("XNAME")"
CacheDirect.P1 = TxtSNAME.Text
CacheDirect.Execute("Do GetList^VISMUTIL(P0,P1,P1,"" """)")
If CDbI(CacheDirect.ErrorRenamed) <> 0 Then
' MsgBox (" Error " & CacheDirect.ErrorName)
Exit Sub
End If
' 取得した名前リストをListBoxに展開
For i = 1 To CacheDirect.NoOfPLISTItem
'UPGRADEWARNING: オブジェクト i の既定プロパティを解決できませんでした。 詳細については、'ms-
help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6A50421D-15FE-4896-8A1B-2EC21E9037B2"
をクリックしてください。
ListLookupName.Items.Add(CacheDirect.PLISTItem(i))
Next i
End Sub
```

ここでCacheDirect.Clear()はラッパークラスが独自に用意していたメソッドで、新しいエミュレータクラスには存在しません。

内容を確認した所、P0-P9,PLIST変数を初期化する処理のようで、処理を俯瞰する限り、ここで初期化する必要がないと判断し、コメントアウトすることにします。

ここでは、ADBKMainのエミュレータクラスを参照する必要があるため、すべてのCacheDirect参照の前にADBKMain.を付加する必要があります。

そして、VB6からVB.NETにコンバートする際に、Errorというキーワードは強制的に変換されるようで、ErrorRenamedになっていたの、これを元のErrorに戻す必要がありました。

NoOfPLISTItemとPLISTItem(i)はラッパークラス独自の実装だったのですが、機能的には全く同じものがエミュレータクラスにあるので、そのメソッドに名前の変更を行いました。

```
Private Sub CmdFindClick(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
Handles CmdFind.Click
```

```
Dim i As Object
'Dim NOL As Short
' 名前のリストを取得
ListLookupName.Items.Clear()
'以下の機能は未実装
'CacheDirect.Clear()
ADBKMain.CacheDirect.PDELIM = Chr(1)
ADBKMain.CacheDirect.P0 = "^ADBK("XNAME")"
ADBKMain.CacheDirect.P1 = TxtSNAME.Text
ADBKMain.CacheDirect.Execute("Do GetList^VISMUTIL(P0,P1,P1,"" """)")
'If CDbI(CacheDirect.ErrorRenamed) <> 0 Then
If CDbI(ADBKMain.CacheDirect.Error) <> 0 Then
MsgBox (" Error " & ADBKMain.CacheDirect.ErrorName)
Exit Sub
End If
' 取得した名前リストをListBoxに展開
```

```

'For i = 1 To CacheDirect.NoOfPLISTItem
  For i = 1 To ADBKMain.CacheDirect.getPLISTLength()
    'UPGRADEWARNING: オブジェクト i の既定プロパティを解決できませんでした。 詳細については、'ms-
    help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6A50421D-15FE-4896-8A1B-2EC21E9037B2"
    をクリックしてください。
    'ListLookupName.Items.Add(CacheDirect.PLISTItem(i))
    ListLookupName.Items.Add(ADBKMain.CacheDirect.getPLIST(i))
  Next i
End Sub

```

次にイベント処理の変更です。

VisM.OCXでExecuteメソッドが実行されたことを補足するためのイベント処理が以下の様に定義されていました。

```

'Private Sub VisM1Executed(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
Handles VisM1.Executed
  Dim status As Object
  If VisM1.P9 <> "" Then
    'UPGRADEWARNING: オブジェクト status の既定プロパティを解決できませんでした。 詳細については、'ms-
    help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6A50421D-15FE-4896-8A1B-2EC21E9037B2"
    をクリックしてください。
    status = MsgBox("データエラー " & VisM1.P9, MsgBoxStyle.OKOnly, "データエラー")
  End If
End Sub

```

Handle以下をエミュレーターのイベントに書き換える必要があります。

ここではVisM.OCXを直接参照していたので、その部分をエミュレータークラスを参照する様に変更します。

```

Private Sub CacheDirectExecuted(ByVal eventSender As System.Object, ByVal eventArgs As
System.EventArgs) Handles CacheDirect.ExecuteEvent
  Dim status As Object
  If CacheDirect.P9 <> "" Then
    'UPGRADEWARNING: オブジェクト status の既定プロパティを解決できませんでした。 詳細については、'
    ms-
    help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6A50421D-15FE-4896-8A1B-2EC21E9037B2"
    をクリックしてください。
    status = MsgBox("データエラー " & CacheDirect.P9, MsgBoxStyle.OkOnly, "データエラー")
  End If
End Sub

```

Errorイベントに関しても同様の変更を行います。

```

'Private Sub VisM1OnError(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
Handles VisM1.OnError
  Dim status As Object
  'UPGRADEWARNING: オブジェクト status の既定プロパティを解決できませんでした。 詳細については、'ms-
  help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6A50421D-15FE-4896-8A1B-2EC21E9037B2"
  をクリックしてください。
  status = MsgBox("Cache' Direct Error " & VisM1.ErrorName, MsgBoxStyle.OKOnly, "Cache' Direct Error")

```

```
'エラーが発生した時点でのローカル変数をダンプする
CacheDirect.PrintLocalVariable()
End Sub
```

```
Private Sub CacheDirectOnError(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
Handles CacheDirect.ErrorEvent
    Dim status As Object
    ' 'UPGRADEWARNING: オブジェクト status の既定プロパティを解決できませんでした。 詳細については、'
ms-
help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6A50421D-15FE-4896-8A1B-2EC21E9037B2"
をクリックしてください。
    status = MsgBox("Cache' Direct Error " & CacheDirect.ErrorName, MsgBoxStyle.OkOnly, "Cache' Direct Error")
    'エラーが発生した時点でのローカル変数をダンプする
    'CacheDirect.PrintLocalVariable()
    MsgBox("エラーが発生しました" & CacheDirect.ErrorName, MsgBoxStyle.OkOnly)
End Sub
```

次に検索した結果で検索するためのOKボタンが押された時の処理です

```
Private Sub CmdOK_Click(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
Handles CmdOK.Click
    Dim result() As String
    CacheDirect.Clear()
    CacheDirect.P0 = VB6.GetItemString(ListLookupName, ListLookupName.SelectedIndex)
    '選択されている名前に対応するデータを検索する。
    CacheDirect.Execute(("Do GetData^ADBK(P0)"))
    'If CDBl(CacheDirect.ErrorRenamed) <> 0 Then
Exit Sub
End If
'検索フォームを非表示
Me.Hide()
'取得データをフォームフィールドに展開する
'CTYPE(ADBKMain.Controls("TxtNAME"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 1)
'CTYPE(ADBKMain.Controls("TxtZIP"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 2)
'CTYPE(ADBKMain.Controls("TxtADDRESS"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 3)
'CTYPE(ADBKMain.Controls("TxtTELH"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 4)
'CTYPE(ADBKMain.Controls("TxtTELO"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 5)
'CTYPE(ADBKMain.Controls("TxtAGE"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 6)
'CTYPE(ADBKMain.Controls("TxtDOB"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 7)
End Sub
```

ここでもClearメソッドが呼ばれていますが、不要と判断し、コメントアウトしました。  
先ほどと同様ErrorがErrorRenameに強制的に変換されていました。  
VB6.GetItemStringもVB6固有のメソッドになるので、書き換えが必要でした。

次にMPieceメソッドもラッパークラス固有の関数でしたが、内容を確認すると.NetのSplit関数を使うともっとシンプルに変換できることがわかりました。

先ほどと同様に先頭にADBKMain.を付加する必要があります。

```
Private Sub CmdOK_Click(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
```

---

```
Handles CmdOK.Click
Dim result() As String
'CacheDirect.Clear()
'CacheDirect.P0 = VB6.GetItemString(ListLookupName, ListLookupName.SelectedIndex)
If (ListLookupName.SelectedIndex >= 0) Then
ADBKMain.CacheDirect.P0 = ListLookupName.Items(ListLookupName.SelectedIndex).ToString()
'選択されている名前に対応するデータを検索する。
ADBKMain.CacheDirect.Execute(("Do GetData^ADBK(P0)"))
'If CDBl(CacheDirect.ErrorRenamed) <> 0 Then
Exit Sub
End If
'検索フォームを非表示
Me.Hide()
'取得データをフォームフィールドに展開する
'CTYPE(ADBKMain.Controls("TxtNAME"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 1)
'CTYPE(ADBKMain.Controls("TxtZIP"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 2)
'CTYPE(ADBKMain.Controls("TxtADDRESS"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 3)
'CTYPE(ADBKMain.Controls("TxtTELH"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 4)
'CTYPE(ADBKMain.Controls("TxtTELO"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 5)
'CTYPE(ADBKMain.Controls("TxtAGE"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 6)
'CTYPE(ADBKMain.Controls("TxtDOB"), Object).Text = CacheDirect.MPiece(CacheDirect.P1, Chr(2), 7)
result = ADBKMain.CacheDirect.P1.ToString().Split(Chr(2))
CTYPE(ADBKMain.Controls("TxtNAME"), Object).Text = result(0)
CTYPE(ADBKMain.Controls("TxtZIP"), Object).Text = result(1)
CTYPE(ADBKMain.Controls("TxtADDRESS"), Object).Text = result(2)
CTYPE(ADBKMain.Controls("TxtTELH"), Object).Text = result(3)
CTYPE(ADBKMain.Controls("TxtTELO"), Object).Text = result(4)
CTYPE(ADBKMain.Controls("TxtAGE"), Object).Text = result(5)
CTYPE(ADBKMain.Controls("TxtDOB"), Object).Text = result(6)
End If
End Sub
```

次にロード処理ですが、このフォーム上では何もする必要がないので、VisMコントロールのロード処理をコメントアウトします。

```
Private Sub FindByName_Load(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
Handles MyBase.Load
'CacheDirect.VisM = CType(Controls("VisM1"), Object)
End Sub
```

フォームのクローズ処理では、エミュレータクラスの終了処理は不要なので、その処理をコメントアウトします。

```
Private Sub FindByNameFormClosed(ByVal eventSender As System.Object, ByVal eventArgs As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
'UPGRADENOTE: オブジェクト CacheDirect をガベージ
コレクトするまでこのオブジェクトを破棄することはできません。 詳細については、'ms-
help://MS.VSCC.v90/dvcommoner/local/redirect.htm?keyword="6E35BFF6-CD74-4B09-9689-3E1A43DF8969"
をクリックしてください。
'CacheDirect = Nothing
End Sub
```

これで修正は終了で、アプリケーションも動作するはずです。

autoinstall.basとVmclass.clsは移行の結果不要になったので、プロジェクトから削除します。

## アプリケーション実行

### エミュレータクラスのロード

OpenExchangeのサイトからCacheDirect.Emulator.clsをダウンロードし、それをIRISにロード（USERネームスペース）します。

### アプリケーションルーチン、グローバル、クラスのロード

ADBKサンプルのGitHubからUser/ADBK.cls,ADBK.mac,VISMUTIL.mac,adbkglb.xmlをダウンロードし、c:\tempにコピーします。

アプリケーションの実行を行うと最初に必要なグローバル、クラス、ルーチンのロードを行うはずですが。

## 最後に

全く無修正というわけにはいきませんが、既存の資産を生かしつつ、VisM.OCXを使用したアプリケーションの移行が簡単にできるということをご理解いただき、VisMアプリケーションの移行にチャレンジしていただきたいと思います。

もう一つ、今回はサーバー側の処理は全く修正していない点も強調しておきたいと思います。

ついでに今回のサンプルでも所々実施したように古い処理をリファクタリングするのもいい機会だと思います。

[#.NET #InterSystems IRIS #Open Exchange](#)  
[InterSystems Open Exchange](#)で関連アプリケーションを確認してください

---

#### ソースURL:

<https://jp.community.intersystems.com/post/vismocxcach%C3%A9-direct%E3%82%92%E5%88%A9%E7%94%A8%E3%81%97%E3%81%9F%E3%82%A2%E3%83%97%E3%83%AA%E3%82%B1%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%82%92iris%E3%81%AB%E7%A7%BB%E8%A1%8C%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95>