

## JRA-VAN 競馬ソフト作成体験教室

### Lesson-3 : JV-Data の内容を読み出す

当コーナーでは、Microsoft Visual Basic 2010 Express Edition(以下 VB 2010 と省略)で「JRA-VAN Data Lab.」サービス対応の競馬ソフトを作成していく過程をステップアップ形式で解説していきます。

今回は、実際に JV-Data をダウンロードする仕組みを実装しました。今回は、前回ダウンロードした JV-Data を読み込む処理を盛り込んでみましょう。さらに、JRA-VAN SDK に同梱されている JV-Data 構造体を用いて、読み込んだ JV-Data を「意味のある区切り」ごとに切り出して表示してみましょう。

#### 【 今回の目標 】

ダウンロードした JV-Data を読み出す。

読み出した JV-Data を「意味のある区切り」で切り出し、表示する。

具体的には、取得した JV-Data のうち、レース詳細 (レコード種別 ID 「RA」) 情報の一部 (開催年、開催月日、競馬場コード、開催回、開催日、レース番号、競走名略称 10 文字) を RichTextBox に表示します。

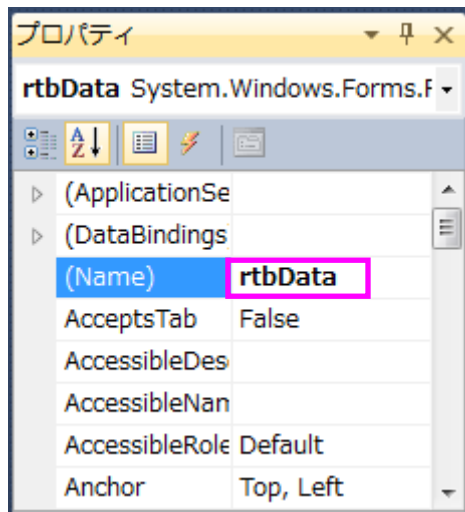
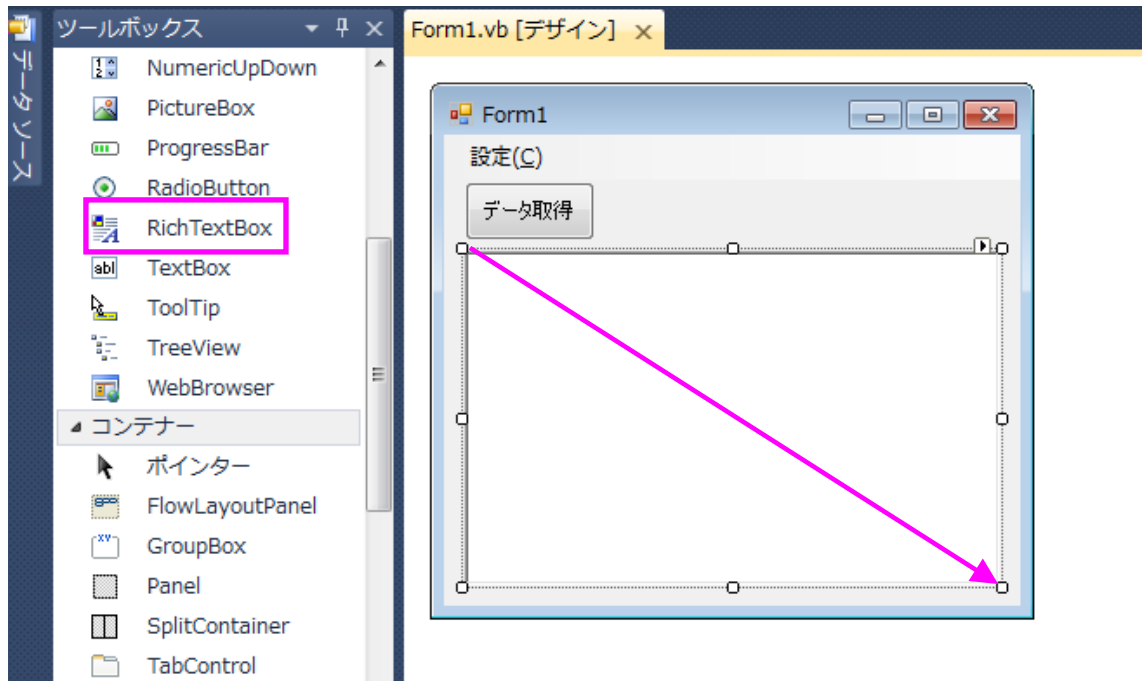
JV-Link に実装されている以下のメソッドの使い方を理解する。

- ・JVRead メソッドで取得した JV-Data を読み出す

#### 【 やってみよう 】

- ① Lesson-2 までを実装したフォーム(frmMain)を含むプロジェクトを開きます。  
(前回のレッスンの続きから開始するのであれば、この作業は必要ありません。)

② フォームにテキスト表示エリア(RichTextBox)を追加します。



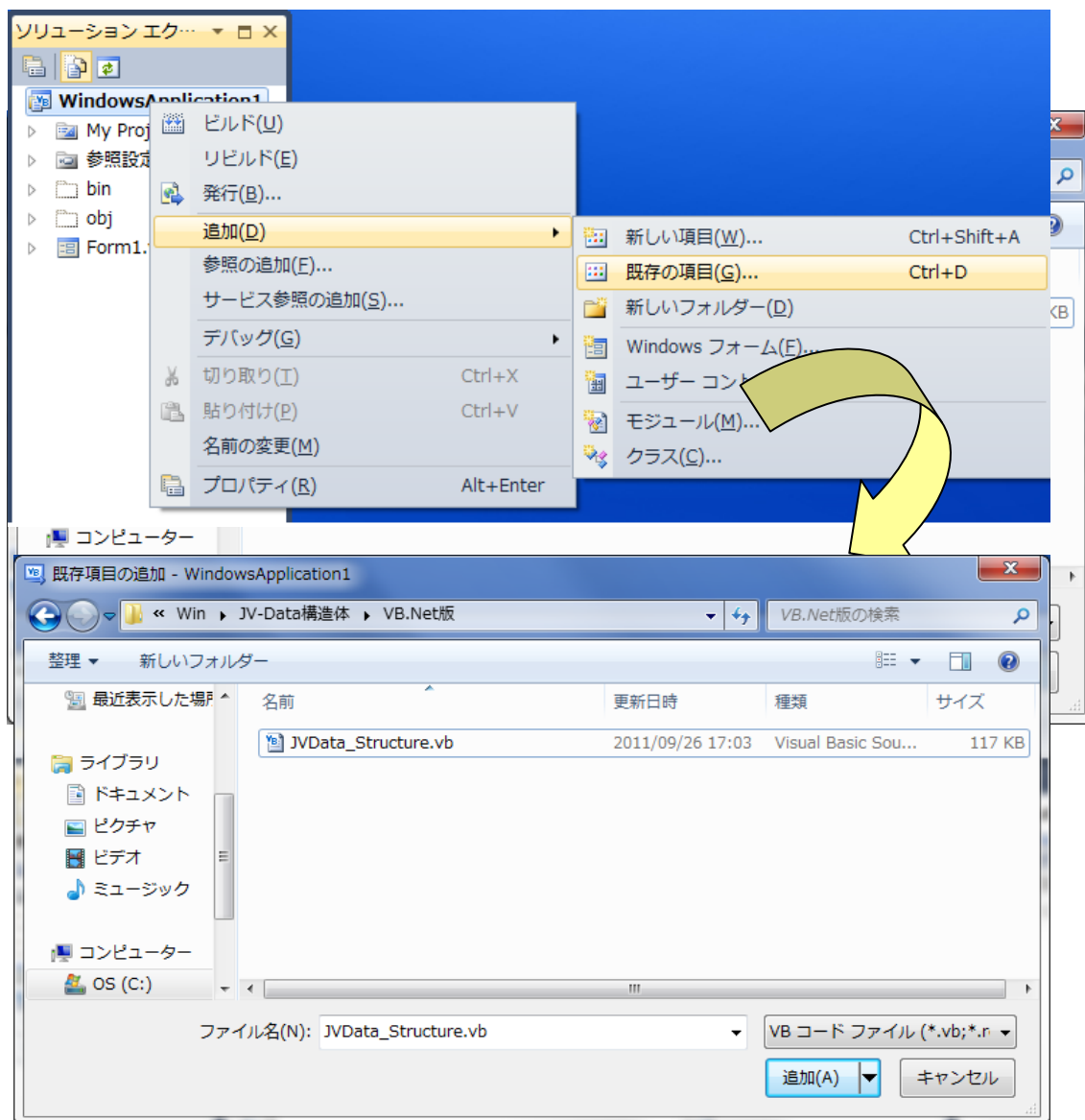
Multiline	True
ReadOnly	False
RightMargin	0
RightToLeft	No
ScrollBars	Both
ShortcutsEnabled	True
ShowSelectionMargin	False
Size	308, 192
TabIndex	3
TabStop	True
Tag	
Text	
UseWaitCursor	False
Visible	True
WordWrap	False
ZoomFactor	1

追加した RichTextBox の各プロパティを以下のように変更します。

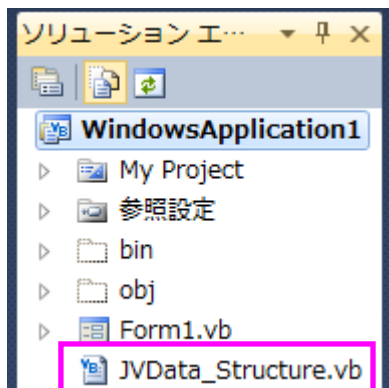
「(Name)」 : rtbData  
 「Multiline」 : True  
 「ScrollBars」 : Both  
 「Text」 : (空欄)  
 「WordWrap」 : False

- ③ 読み出した **JV-Data** は1レコード毎に連続した固定長の文字列となっています。この文字列を「意味のある区切り」ごとに分割するためには、**JV-Data** 仕様書を参考に、「○  
○バイト目から●●バイト分は◇◇情報」のように順次切り出していく必要があります。今回は、**JRA-VAN SDK** に同梱されている「**JV-Data** 構造体」(正確には構造体+構造体への格納関数)を用いて、複雑なコーディングをすることなく、簡単に「意味のある区切り」ごとに切り出してみましょう。

ソリューションエクスプローラーより右クリックにて「追加(D)」→「既存の項目(G)」を選択してダイアログを表示し、「**JRA-VAN SDK**」の解凍後ディレクトリの下「¥JV-Data 構造体¥VB.Net 版」にある「JVData\_Structure.vb」を選択して、「開く」をクリックします。



- ④ ソリューションエクスプローラーに「JVData\_Structure.vb」が追加されます。



- ⑤ レース詳細情報の一部をリッチテキストボックスに表示するように「データ取得」ボタンクリック時の処理を修正しましょう（灰色部分は前回コーディング済）。

```
Private Sub btnGetJVData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetJVData.Click

    Dim IReturnCode As Long

    Try
        Dim strDataSpec As String          ' 引数 JVOpen:ファイル識別子
        Dim strFromTime As String          ' 引数 JVOpen:データ提供日付
        Dim IOption As Long                ' 引数 JVOpen:オプション
        Dim IReadCount As Long             ' JVLink 戻り値
        Dim IDownloadCount As Long         ' JVOpen:総ダウンロードファイル数
        Dim strLastFileTimestamp As String ' JVOpen:最新ファイルのタイムスタンプ

        Const IBufferSize As Long = 110000 ' JVRead:データ格納バッファサイズ
        Const INameSize As Integer = 256   ' JVRead:ファイル名サイズ
        Dim strBuff As String              ' JVRead:データ格納バッファ
        Dim strFileName As String           ' JVRead:ダウンロードファイル名
        Dim RaceInfo As JV_RA_RACE         ' レース詳細情報構造体

        ' 引数設定
        strDataSpec = "RACE"
        strFromTime = "0000000000000000"
        IOption = "2"

        ' JVLinkダウンロード処理
        IReturnCode = Me.AxJVLink1.JVOpen(strDataSpec, strFromTime, IOption,
            IReadCount, IDownloadCount, strLastFileTimestamp)

        ' エラー判定
        If IReturnCode <> 0 Then
            MsgBox("JVOpenエラー:" & IReturnCode)
        Else
            MsgBox("戻り値:" & IReturnCode & vbCrLf &
                "読み込みファイル数:" & IReadCount & vbCrLf &
                "ダウンロードファイル数:" & IDownloadCount & vbCrLf &
                "タイムスタンプ:" & strLastFileTimestamp)
        End If
    End Try
End Sub
```

Point1

```

If IReadCount > 0 Then
  Do
    'バッファ作成
    strBuff = New String(vbNullChar, IBufferSize)
    strFileName = New String(vbNullChar, INameSize)

    'JVRead で 1 行読み込み
    IReturnCode = Me.AxJVLink1.JVRead(strBuff, IBufferSize, _
      strFileName)

    'リターンコードにより処理を分枝
    Select Case IReturnCode
      Case 0 '全ファイル読み込み終了
        Exit Do
      Case -1 'ファイル切り替わり
      Case -3 'ダウンロード中
      Case -201 'Init されてない
        MsgBox("JVInit が行われていません。")
        Exit Do
      Case -203 'Open されてない
        MsgBox("JVOpen が行われていません。")
        Exit Do
      Case -503 'ファイルがない
        MsgBox(strFileName & "が存在しません。")
        Exit Do
      Case Is > 0 '正常読み込み
        'レコード種別 ID の識別
        If Mid(strBuff, 1, 2) = "RA" Then
          'レース詳細のみ処理
          'レース詳細構造体への展開
          RaceInfo.SetData(strBuff)

          'データ表示
          rtbData.AppendText( _
            "年:" & RaceInfo.id.Year & _
            "月日:" & RaceInfo.id.MonthDay & _
            "場:" & RaceInfo.id.JyoCD & _
            "回次:" & RaceInfo.id.Kaiji & _
            "日次:" & RaceInfo.id.Nichiji & _
            "R:" & RaceInfo.id.RaceNum & _
            "レース名:" & RaceInfo.RaceInfo.Ryakusyo10 _
            & vbCrLf)
        End If
      End Select
    Loop While (1)
  End If
End If

Catch
  Debug.WriteLine(Err.Description)
  Exit Sub

End Try

'JVLink 終了処理
IReturnCode = Me.AxJVLink1.JVClose()
If IReturnCode <> 0 Then
  MsgBox("JVClose エラー:" & IReturnCode)
End If

End Sub

```

[ソースコード 005-01]

• Point1

③～④で組み込んだ「JV-Data 構造体」で定義されている「レース詳細情報構造体(JV\_RA\_RACE)」型の構造体「RaceInfo」を宣言。

• Point2

Do～Loop While 内で JVRead を繰り返し、JV-Data を 1 レコードずつ読み込み「Buff」に格納。

• Point3

Select Case を用いて JVRead の戻り値に応じた処理を実行 (JVRead の戻り値に関しては、「JV-Link インターフェース仕様書」の「3. コード表」を参照)。

• Point4

レコード種別 ID が「RA」(レース詳細) のレコードのみ処理を実行。

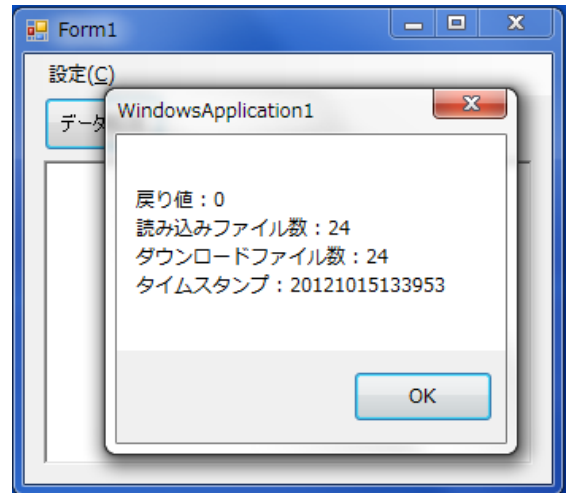
• Point5

「Buff」に格納されたレコードを構造体「RaceInfo」に展開。

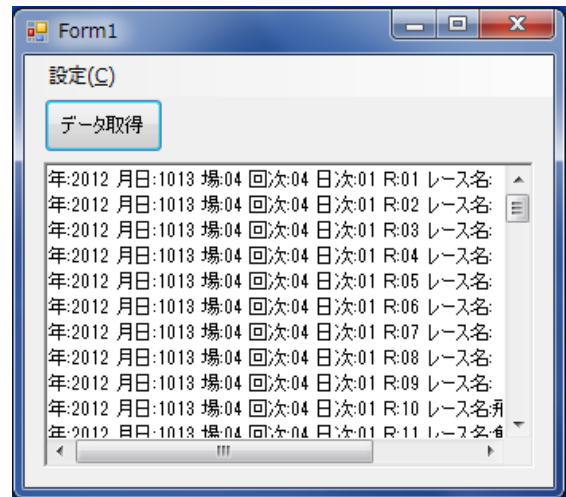
### 【 確認しよう 】

それでは、実際に動かしてみましょう。

- ① メニューから「デバッグ」→「デバッグ開始」を選択し、プログラムを実行します。エラーが発生した場合は、タスク一覧にエラーメッセージが表示されるので、正しく修正してエラーが無くなるまでビルドします。前回までに「サービスキー」を設定していない場合は、フォーム上のメニューから「設定」→「JV-Link の設定」を選択して「JV-Link 設定」ダイアログを表示し、「サービスキー」を設定します。
- ② フォームの「データ取得」ボタンをクリックすると、前回実装したコーディングにより、メッセージボックスがポップアップします。メッセージボックスには、リターンコードや読み込みファイル数、ダウンロードファイル数、最新読み込みファイルのタイムスタンプが表示されます。これらの数値は、実行するタイミングによって異なります。詳しくは、Lesson-2 をご覧下さい。



- ③ メッセージボックスを「OK」で閉じるとレース詳細情報のうち「開催年、開催月日、競馬場コード、開催回、開催日目、レース番号、競走名略称 10 文字」が RichTextBox に表示されます。表示するレコード種別や項目などを変更して色々試してみましょう。



### [ ワンポイントメモ ]

実際に競馬ソフトを作成する際には、取得した情報をデータベースやテキストファイルなどに格納しておきます。