

JRA-VAN 競馬ソフト作成体験教室

Lesson-4 : JV-Data のダウンロード/読み込み進捗状況を表示する

当コーナーでは、Microsoft Visual Basic 2019 Professional Edition(以下 VB 2019 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。

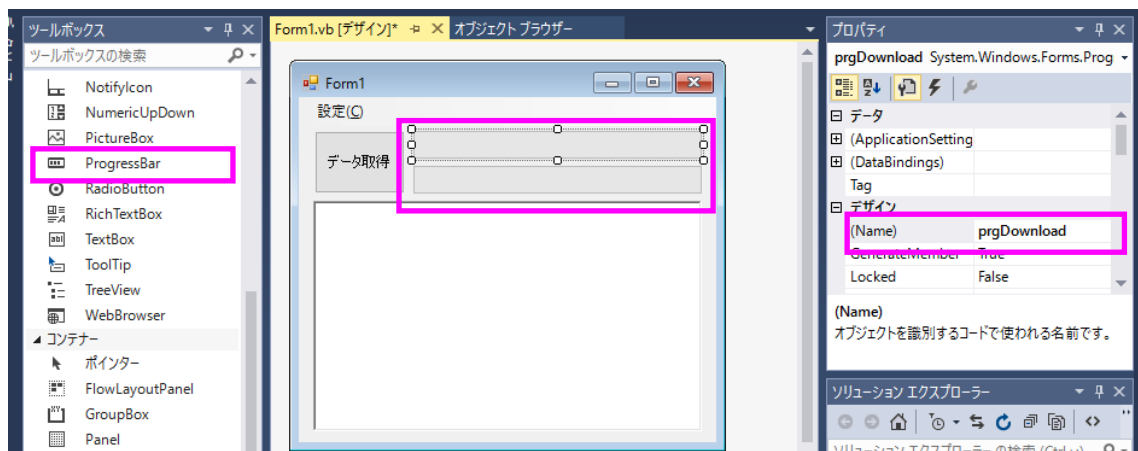
前回までで、実際に JV-Data をダウンロードして内容を表示する簡単な仕組みを実装しました。今回は、プログレスバーを追加して、JV-Data のダウンロード/読み込み処理の進捗情報を表示できるようにしてみましょう。

【 今回の目標 】

- ・プログレスバーを追加して JV-Data のダウンロード/読み込み進捗状況を表示する。
〔 具体的には、プログレスバーを2本用意して、JV-Data のダウンロード進捗状況と読み込み進捗状況をそれぞれ表示する。 〕
- ・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。
JVStatus メソッドで JV-Data のダウンロード進捗情報を取得する。

【 やってみよう 】

- ① Lesson-3 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 フォームにプログレスバー(ProgressBar)を2つ追加し、各 ProgressBar のプロパティを以下のように変更します。
上段の ProgressBar のデザイナー 「(Name)」 : prgDownload
下段の ProgressBar のデザイナー 「(Name)」 : prgJVRead



[ワンポイントメモ]

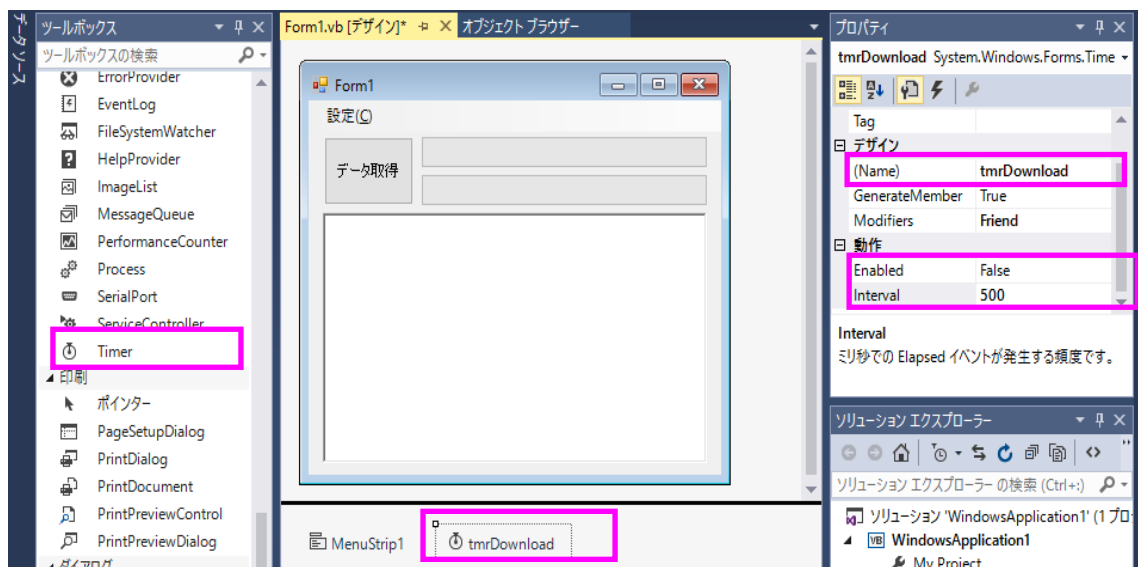
上段のプログレスバーでは **JV-Data** のダウンロードの進捗状況を、下段のプログレスバーでは **JV-Data** の読み込みの進捗状況をそれぞれ表示します。

③ フォームにタイマー(Timer)を追加し、プロパティを以下のように変更します。

デザイナー「(Name)」 : tmrDownload

動作「Enabled」 : False

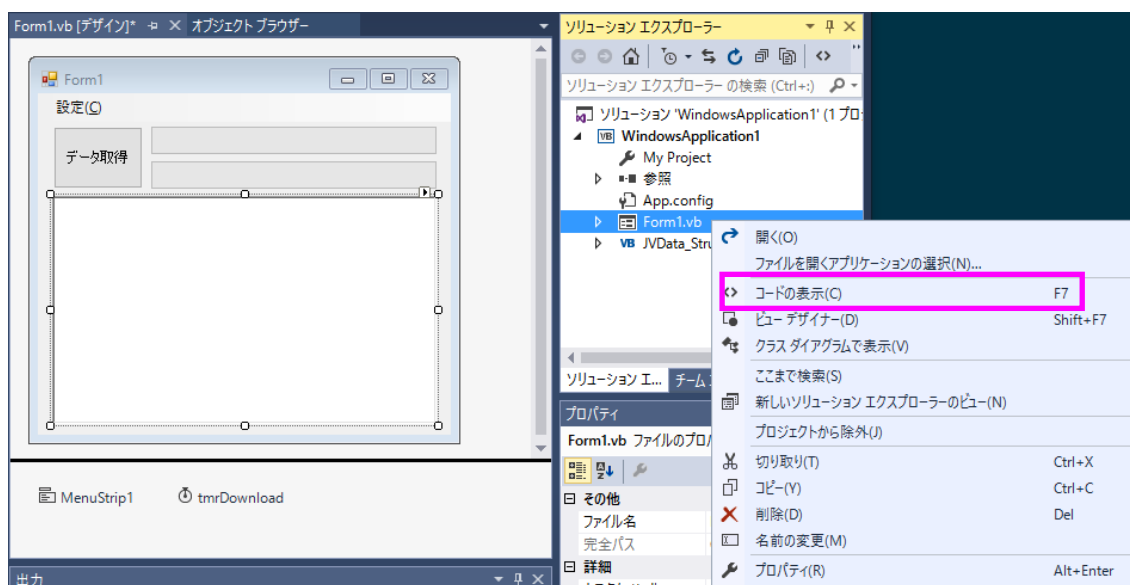
動作「Interval」 : 500



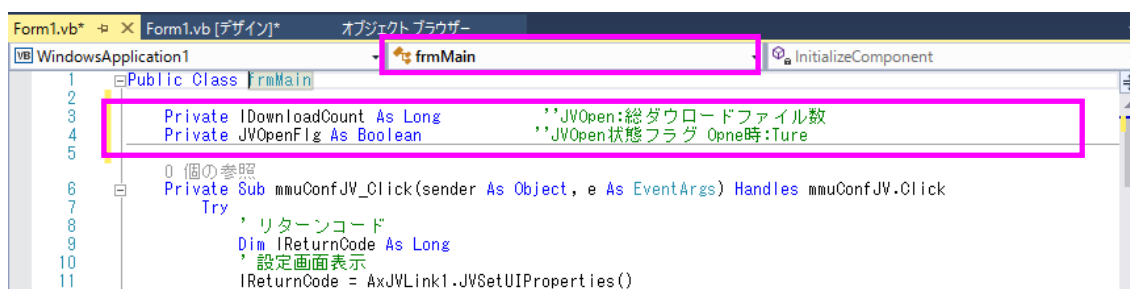
[ワンポイントメモ]

JV-Data のダウンロードは JVOpen 後、バックグラウンドで実行されるので、タイマーを用いて定期的に進捗状況をチェックします。

- ④ フォーム「frmMain」のコードエディタウィンドウを表示します。
(ソリューションエクスプローラの「From1.vb」右クリックし、メニューの中から「コードの表示」をクリックします。



- ⑤ コードエディタウィンドウ上部にある左側リストボックスから「frmMain」を選択後、以下のソースコードを入力します。



追加するコード

```
Private IDownloadCount As Long          '' JVOpen:総ダウンロードファイル数
Private JVOpenFlg As Boolean            '' JVOpen 状態フラグ Opne 時:Ture
```

[ソースコード 006-01]

- ⑥ コードエディタウィンドウで「データ取得」ボタンクリック時の処理を修正します（灰色部分は前回コーディング済）。 [ソースコード 006-02]

```
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 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

        ' 進捗表示初期設定
        tmrDownload.Enabled = False      '' タイマー停止
        prgDownload.Value = 0            '' JVData ダウンロード進捗
        prgJVRead.Value = 0              '' JVData 読み込み進捗

        ' 引数設定
        strDataSpec = "RACE"
        strFromTime = "20050301000000"
        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)

            ' 進捗表示プログレスバー最大値設定
            If IDownloadCount = 0 Then
                prgDownload.Maximum = 100 '' ダウンロード必要無し
                prgDownload.Value = 100
            Else
                prgDownload.Maximum = IDownloadCount
                tmrDownload.Enabled = True '' タイマー開始
            End If
            prgJVRead.Maximum = IReadCount

            ' バックグラウンドでの処理を実行
            Application.DoEvents()

            ' バッファ作成
            strBuff = New String(vbNullChar, IBufferSize)
        End Try
    End Sub
```

Point1

Point2

```

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
    prgJVRead.Value = prgJVRead.Maximum ' 進捗表示
    Exit Do
Case -1 ' ファイル切り替わり
    prgJVRead.Value = prgJVRead.Value + 1
Case -3 ' ダウンロード中
Case -201 ' Init されていない
    MsgBox("JVInit が行われていません。")
    Exit Do
Case -203 ' Open されていない
    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

' タイマ有効時は、無効化する
If tmrDownload.Enabled = True Then
    tmrDownload.Enabled = False
    prgDownload.Value = prgDownload.Maximum
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

```

Point3

Point4

• Point 1

JV-Data をダウンロードする必要がなければ、ダウンロード進捗状況を 100%にしてしまう。この場合タイマーを動かす必要はない。

• Point 2

ループ毎に DoEvents 処理を実行し、他のイベント（この場合タイマーイベント）処理が行われるための隙間を作ります。

• Point 3

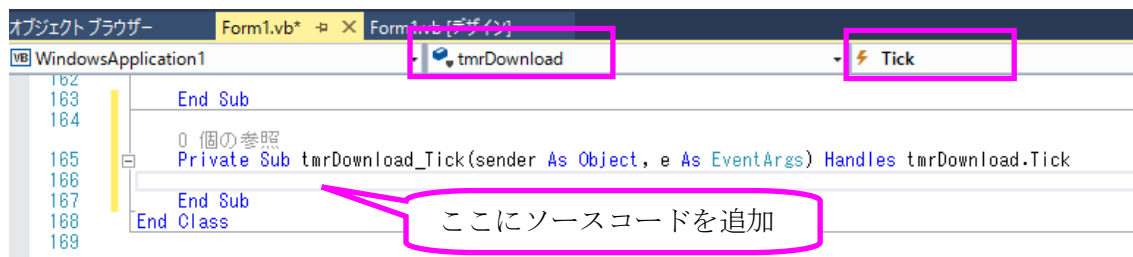
1 ファイル読み込む毎に全読み込みファイル数の対する JV-Data 読み込み進捗状況を表すプログレスバーを進めます。

• Point 4

JV-Data のダウンロード処理の進捗状況が 100%になる前に JV-Data の読み込み処理の進捗状況が 100%になってしまった場合のための処理を記述します。これを記述しておかないと、タイマー間隔を長く設定した際にエラーが発生する可能性があります。

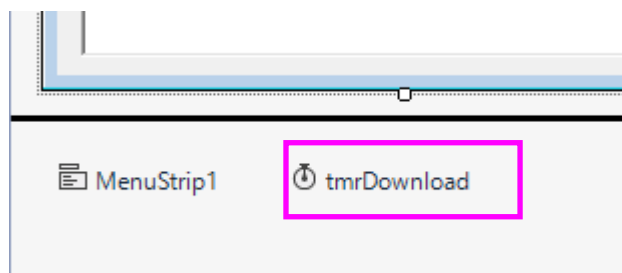
なお、Lesson-3 でプロシージャ「btnGetJVData_Click ()」の中で宣言した IDownloadCount は削除しておきます。

コードエディタウィンドウの左側リストボックスから「tmrDownload」を、右側リストボックスから「Tick」を選択すると、tmrDownload のメソッドが追加されるので、以下の[ソースコード 006-03]を追加します。



[ワンポイントメモ]

フォームの下の「tmrDownload」アイコンをダブルクリックしても tmrDownload のメソッドが追加されます。



```
Private Sub tmrDownload_Tick(ByVal sender As Object, ByVal e As System.EventArgs)
Handles tmrDownload.Tick

    Dim IReturnCode As Long                '' JVLink返回值

    ' JVLinkダウンロード進捗率
    IReturnCode = AxJVLink1.JVStatus      '' ダウンロード済のファイル数を返す

    ' エラー判定
    If IReturnCode < 0 Then
        ' エラー
        MsgBox("JVStatusエラー:" & IReturnCode)
        ' タイマー停止
        tmrDownload.Enabled = False

        ' JVLink終了処理
        IReturnCode = Me.AxJVLink1.JVClose()
        If IReturnCode <> 0 Then
            MsgBox("JVCloseエラー:" & IReturnCode)
        End If
    ElseIf IReturnCode < IDownloadCount Then
        ' ダウンロード中
        ' プログレス表示
        prgDownload.Value = IReturnCode
    ElseIf IReturnCode = IDownloadCount Then
        ' ダウンロード完了
        ' タイマー停止
        tmrDownload.Enabled = False
        ' プログレス表示
        prgDownload.Value = IReturnCode
    End If

End Sub
```

[ソースコード 006-03]

③でタイマーを追加したときに、「Interval」に 500 を設定したので、タイマーがスタートしてから 500 ms ごとにこのメソッドが呼び出されます。

• Point 1

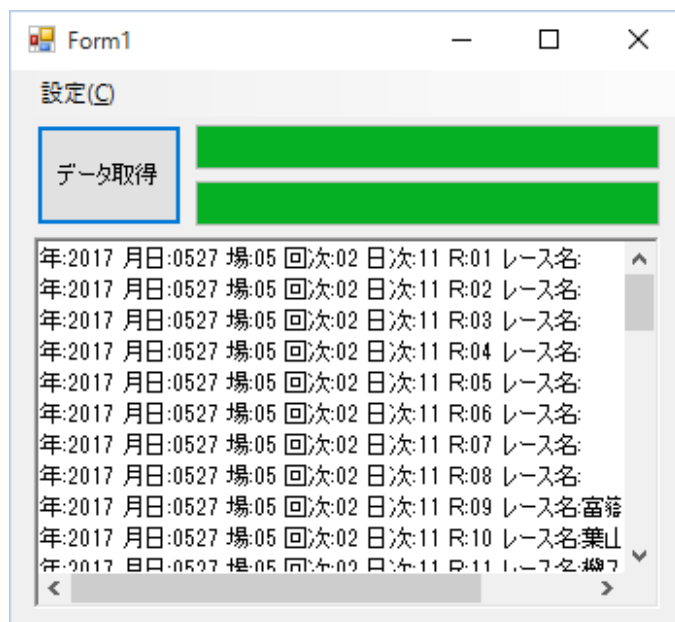
ダウンロード済みのファイル数を取得してダウンロード用のプログレスバーの Value プロパティに設定することにより、プログレスバーの表示を更新します。

【 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。
- ② JV-Link を使用するための「利用キー」を設定していない場合は、メインフォーム上のメニューから「設定」→「JV-Link の設定」を選択し、「利用キー」を設定します（詳細は Lesson-1 を参照）。
- ③ フォームの「データ取得」ボタンをクリックすると、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。
- ④ メッセージボックスを「OK」で閉じると、前回同様、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、レース番号、競走名略称 10 文字」が表示されます。

JV-Data のダウンロード進捗状況や読み込み進捗状況は上下のプログレスバーで確認することができます。



[ワンポイントメモ]

Lesson-2 を参考にダウンロードすべきファイルが「cache」フォルダに存在する場合と存在しない場合で、プログレスバーの動きの違いを比べてみましょう。