

Ultra Light.NET を使用した同期プログレス・バーの実装

多くの場合、アプリケーションを同期しているときにはステータス・プログレス・バーをユーザに表示すると便利です。このマニュアルでは、VB.NET を使用して .NET 内の Ultra Light アプリケーションに対するプログレス・バーを実装する方法について説明します。

必要なソフトウェア

1. SQL Anywhere 11 と Ultra Light
2. Ultra Light サンプル・データベース (custdb)
3. Microsoft Visual Studio 2008

目標

このマニュアルでは、既存の CustDB Sync ダイアログを変更してプログレス・バーを追加する方法について説明します。これにより、同期にかかる時間について、より正確な情報がユーザに提供されます。プログレス・バーを追加するには、以下の手順を実行します。

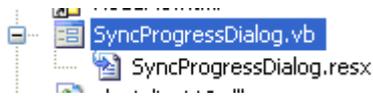
1. Visual Studio 2008 に CustDB プロジェクトをロードする
2. プロジェクトにプログレス・バーを追加する
3. プログレス・バーを実装するためのコードを追加する
4. Mobile Link を実行する
5. アプリケーションを配備する
6. 同期する

ステップ 1： Visual Studio 2008 に CustDB ソリューションをロードする

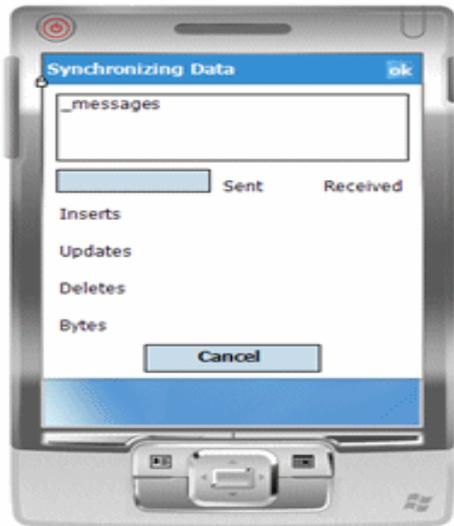
1. SQL Anywhere 11 samples ディレクトリ内の CustDB サンプルに移動します。デフォルトでは、このサンプルは `C:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples\UltraLite.NET\CustDB\ce\vs8` (Windows XP の場合) と `C:\Users\Public\Documents\SQL Anywhere 11\Samples\UltraLite.Net\CustDB\ce\vs8` (Windows Vista 以降の場合) にインストールされます。
2. `CustDB.sln` ファイルをロードします。プロジェクトまたはソリューションの変換を求めるプロンプトが表示されたら、その変換を行い、すべてのデフォルトを受け入れます。

ステップ 2： プログレス・バーをプロジェクトに追加する

1. [ソリューション エクスプローラ] で、*SyncProgressDialog.vb* ファイルをダブルクリックします。



2. [ツールボックス] からプログレス・バーをドラッグして、[Status Sync] ダイアログの下に配置します。一例として、以下のスクリーンショットを参照してください。



ステップ 3： プログレス・バーを実装するためのコードを追加する

ProgressBar に以下のプロパティを設定します。

- a. [Name] : progress
 - b. [Maximum] : 100
 - c. [Minimum] : 1
 - d. [Value] : 1
2. *SyncProgressDialog.vb* のコードを開いて、以下の関数に移動します。
`Public Overridable Function SyncProgressed(_)`
 3. Select Case の順序が以下のように設定されていることを確認します。プログレス・バーが正しく機能するためには、この順序が非常に重要です。

```
Select Case data.State
    Case ULSyncProgressState.STATE_STARTING
        :
        :
    Case ULSyncProgressState.STATE_CANCELLED
```

```

:
:
Case ULSyncProgressState.STATE_CONNECTING
:
:
Case ULSyncProgressState.STATE_SENDING_HEADER
:
:
Case ULSyncProgressState.STATE_SENDING_TABLE
:
:
Case ULSyncProgressState.STATE_SENDING_DATA
:
:

Case ULSyncProgressState.STATE_FINISHING_UPLOAD
:
:

Case ULSyncProgressState.STATE_RECEIVING_UPLOAD_ACK
:
:

Case ULSyncProgressState.STATE_RECEIVING_TABLE
:
:

Case ULSyncProgressState.STATE_RECEIVING_DATA
:
:

Case ULSyncProgressState.STATE_COMMITTING_DOWNLOAD
:
:

Case ULSyncProgressState.STATE_SENDING_DOWNLOAD_ACK
:
:

Case ULSyncProgressState.STATE_DISCONNECTING
:
:

Case ULSyncProgressState.STATE_DONE
:
:

Case ULSyncProgressState.STATE_ERROR
:
:

End Select
```

4. コードを追加します。アルゴリズムは、以下のように機能します。
 - a. プログレス・バーを進めるために必要なケース数を決定します。この例では、プログレス・バーが進む必要のあるケースが 11 個あり、進まないケースが 3 つあります。最初のケースから開始し、1 ずつインクリメントして、ケースの総数で割ります。たとえば、`Case ULSyncProgressState.STATE_STARTING` では、これは 1/11 です。2 番目の `Case ULSyncProgressState.STATE_CANCELLED` では、プログレス・バーは進みません。`Case ULSyncProgressState.STATE_CONNECTING` では 2/11 となり、それ以降も同様です。
 - b. グローバル変数を作成します (この例では `_percentage`)。たとえば、`Private _percentage As Double` を作成します。この変数は、変数を宣言する以下のセクションで作成することができます。

```

Private _allowClose As Boolean ' Tells OnClosing to proceed
Private _doDialog As Boolean ' Tells ShowDialog to process events
Private _inSync As Boolean ' Used to interpret the Close button
Private _syncCancelled As Boolean
Private _dismiss As Boolean
Private _conn As ULConnection
Private _resources As ResourceManager
Private _syncSQLCode As ULSQLCode Private _cancelString As String
Private _dismissString As String Private WithEvents _cancelAs
System.Windows.Forms.Button
Private _sentLabel As System.Windows.Forms.Label
Private _recLabel As System.Windows.Forms.Label
Private _insertLabel As System.Windows.Forms.Label
Private _updateLabel As System.Windows.Forms.Label
Private _deleteLabel As System.Windows.Forms.Label
Private _bytesLabel As System.Windows.Forms.Label
Private _insertsSent As System.Windows.Forms.Label
Private _updatesSent As System.Windows.Forms.Label
Private _deletesSent As System.Windows.Forms.Label
Private _bytesSent As System.Windows.Forms.Label
Private _insertsRec As System.Windows.Forms.Label
Private _updatesRec As System.Windows.Forms.Label
Private _deletesRec As System.Windows.Forms.Label
Private _bytesRec As System.Windows.Forms.Label
Private _message As System.Windows.Forms.TextBox
Friend WithEvents progress As System.Windows.Forms.ProgressBar
Private WithEvents _timer As System.Windows.Forms.Timer
Private _percentage As Double 'to support progressbar

```

5. グローバル変数を作成したら、以下の太字の行を既存の `select case` コードに追加します。


```

Select Case data.State
    Case ULSyncProgressState.STATE_STARTING
        _percentage = (1 / 11)

```

```
Case ULSyncProgressState.STATE_CANCELLED
    msg = GetMessage(data)

Case ULSyncProgressState.STATE_CONNECTING
    msg = GetMessage(data)
    _percentage = ( 2 / 11)

Case ULSyncProgressState.STATE_SENDING_HEADER
    msg = GetMessage(data)
    _percentage = ( 3 / 11)

Case ULSyncProgressState.STATE_SENDING_TABLE
    msg = GetMessage(data)
    _percentage = ( 4 / 11)

Case ULSyncProgressState.STATE_SENDING_DATA
    _insertsSent.Text = data.SentInserts.ToString()
    _updatesSent.Text = data.SentUpdates.ToString()
    _deletesSent.Text = data.SentDeletes.ToString()

Case ULSyncProgressState.STATE_FINISHING_UPLOAD
    _insertsSent.Text = data.SentInserts.ToString()
    _updatesSent.Text = data.SentUpdates.ToString()
    _deletesSent.Text = data.SentDeletes.ToString()
    _percentage = ( 5 / 11)

Case ULSyncProgressState.STATE_RECEIVING_UPLOAD_ACK
    msg = GetMessage(data)
    _percentage = ( 6 / 11)

Case ULSyncProgressState.STATE_RECEIVING_TABLE
    msg = GetMessage(data)
    _percentage = ( 7 / 11)

Case ULSyncProgressState.STATE_RECEIVING_DATA
    _insertsRec.Text = data.ReceivedInserts.ToString()
    _updatesRec.Text = data.ReceivedUpdates.ToString()
    _deletesRec.Text = data.ReceivedDeletes.ToString()

Case ULSyncProgressState.STATE_COMMITTING_DOWNLOAD
    _insertsRec.Text = data.ReceivedInserts.ToString()
    _updatesRec.Text = data.ReceivedUpdates.ToString()
    _deletesRec.Text = data.ReceivedDeletes.ToString()
    _percentage = ( 8 / 11)

Case ULSyncProgressState.STATE_SENDING_DOWNLOAD_ACK
    msg = GetMessage(data)
    _percentage = ( 9 / 11)

Case ULSyncProgressState.STATE_DISCONNECTING
    msg = GetMessage(data)
    _percentage = (10 / 11)

Case ULSyncProgressState.STATE_DONE
    msg = GetMessage(data)
```

```

        _percentage = (11 / 11)
    Case ULSyncProgressState.STATE_ERROR
        msg = GetErrorMessage(data.SQLCode, data.ErrorMessage,
data.SyncResult.UploadOK)
        ' Change Cancel button to dismiss
        _cancel.Text = _dismissString
        _dismiss = True
    End Select
    If Not msg Is Nothing Then
        _message.Text = msg
    End If
    If data.State <> ULSyncProgressState.STATE_LAST Then
        _bytesSent.Text = data.SentBytes.ToString()
        _bytesRec.Text = data.ReceivedBytes.ToString()
    End If
    Application.DoEvents()
    ' Update the progress bar value
    progress.Value = CInt(100 * (_percentage))
    Return _syncCancelled

```

6. プロジェクトを保存します。
7. プロジェクトをビルドします。
8. 構文エラーをすべて修正します。

注：この例では、パーセンテージを計算する必要のないケースが 3 つあります。それらのケースを以下に示します。

1. ULSyncProgressState.STATE_CANCELLED
2. ULSyncProgressState.STATE_SENDING_DATA
3. ULSyncProgressState.STATE_RECEIVING_DATA

最初のケースでは、ユーザがキャンセルした場合、何の計算も行いません。残りの 2 つのケースでは、各ケースがテーブルごとに起動されます。たとえば、テーブルごとのデータが統合データベースに送信されると、STATE_SENDING_DATA が起動されます。また、テーブルごとのデータが統合データベースから受信されると、STATE_RECEIVING_DATA が起動されます。これらのケースで計算を行うと、プログレス・バーが前後にジャンプします（つまり、数値が上下する）。そのため、この 2 つのケースでは何も行わないことをおすすめします。

ステップ 4： Mobile Link を実行する

これで Mobile Link サーバを実行できるようになりました。

1. [スタート] メニューから [プログラム] > [SQL Anywhere 11] > [Mobile Link] > [同期サーバーのサンプル] を選択します。
Mobile Link サーバが起動し、新しい要求を受け入れるために待機します。

ステップ 5： アプリケーションを配備する

1. Visual Studio で、[F5] を押します。
エミュレータまたは物理デバイスにソリューションを配備できます。
2. アプリケーションが初めて配備されると、ユーザ ID の入力を求めるダイアログが表示されます。デフォルト (ユーザ ID 50) を受け入れます。

ステップ 6： 同期する

デフォルト画面で [OK] をクリックすると、アプリケーションが自動的に同期します。ただし、手動で同期を開始することもできます。

1. アプリケーションが実行されているデバイスまたはエミュレータで、[File] をクリックします。
2. [Synchronize] をクリックします。