How to read a SQLite database into a DataGridView object

I am writing a VB program using VS 2013. I am using the methods in System.Data.SqLite.dll from SQLite.org. I can read my database fine into a ListBox object. I am posting my code that I am using for this. What I would like to do is send this data to a DataGridView object. I am having no luck doing it correctly.

```vbnet
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    Dim f As New OpenFileDialog
    f.Filter = "SQLite 3 (*.db)|*.db|All Files|*.*"
    If f.ShowDialog() = DialogResult.OK Then
        Dim SQLconnect As New SQLite.SQLiteConnection()
        Dim SQLcommand As SQLiteCommand
        SQLconnect.ConnectionString = "Data Source=" & f.FileName & ";"
        SQLconnect.Open()
        SQLcommand = SQLconnect.CreateCommand
        SQLcommand.CommandText = "SELECT address, date, body FROM sms ORDER BY date DESC"
        Dim SQLreader As SQLiteDataReader = SQLcommand.ExecuteReader()

        lst_records.Items.Clear()

        While SQLreader.Read()
            lst_records.Items.Add(String.Format("address = {0}, date = {1}, body = {2}",
SQLreader(0), SQLreader(1), SQLreader(2)))
        End While

        SQLcommand.Dispose()
        SQLconnect.Close()
    End If
End Sub
```

vb.net   sqlite   system.data.sqlite

5 Answers

I found a few similar questions on StackOverflow, but not close enough to post. Sorry to send you to another website. http://cplus.about.com/od/howtodothingsinc/ss/How-To-Use-Sqlite-From-Csharp_2.htm This is a copy/paste from the above link. The answer is to use the SQLLiteConnection and SQLLiteDataAdapter. Below is in C#, but easily convertable to VB.

```csharp
private void btngo_Click(object sender, EventArgs e)
 {
    const string filename = @"C:\cplus\tutorials\c#\SQLite\About.sqlite";
    const string sql = "select * from friends;";
```

```
      var conn = new SQLiteConnection("Data Source=" + filename + ";Version=3;") ;
      try
      {
        conn.Open() ;
        DataSet ds = new DataSet() ;
        var da = new SQLiteDataAdapter(sql, conn) ;
        da.Fill(ds) ;
        grid.DataSource = ds.Tables[0].DefaultView;
      }
      catch (Exception)
      {
  throw;
      }
  }
```

answered Apr 24 '14 at 19:25
user3525511

Using Microsoft SQL you would fill the result to a DataTable by passing it to an instance of SQLDataAdapter. Once the data table is filled, you can assign it to the DataGridView as a data source. The benefit of this is that the DataGridView can automatically update its content.

I have no experience with using SQL Lite, but I expect that its API is similar.

```
Dim table As New DataTable()
Dim dataAdapter As New SqlClient.SqlDataAdapter(cmd)

dataAdapter.Fill(table)
dataGridView.DataSource = table
```

answered Apr 24 '14 at 18:57
Dennis Kassel
1,078 2 9 21

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    Dim f As New OpenFileDialog
    f.Filter = "SQLite 3 (*.db)|*.db|All Files|*.*"
    If f.ShowDialog() <> DialogResult.OK Then Exit Sub

    lst_records.Items.Clear()
    Using SQLconnect As New SQLiteConnection("Data Source=" & f.FileName & ";"), _
          SQLcommand As New SQLiteCommand("SELECT address, date, body FROM sms ORDER BY
date DESC")

        SQLconnect.Open()
        Using SQLreader As SQLiteDataReader = SQLcommand.ExecuteReader()
            MyDataGridView.DataSource = SQLreader
        End Using
    End Using
End Sub
```

answered Apr 25 '14 at 13:34
Joel Coehoorn
262k 92 455 677

```vbnet
Function getData(ByVal sql As String) As DataTable

    OpenConnection()
    SQLcommand = SQLconnect.CreateCommand
    SQLcommand.CommandText = sql
    Dim dataAdapter As New SQLiteDataAdapter(SQLcommand)
    Dim table As New DataTable
    dataAdapter.Fill(table)
    CloseConnection()
    Return table
End Function
```

On Main MyDataGrid.DataSource = getData("select * from tb_employee")

edited Mar 25 '16 at 4:26   answered Mar 25 '16 at 4:16

Bagoes
1   1

This works for me:

```vbnet
Dim conn = New SQLiteConnection("Data Source=MyDataBaseName.sqlite;Version=3")

Try
    Using (conn)
        conn.Open()

        Dim sql = "SELECT * FROM table"

        Dim cmdDataGrid As SQLiteCommand = New SQLiteCommand(sql, conn)

        Dim da As New SQLiteDataAdapter
        da.SelectCommand = cmdDataGrid
        Dim dt As New DataTable
        da.Fill(dt)
        DataGridView1.DataSource = dt

        Dim readerDataGrid As SQLiteDataReader = cmdDataGrid.ExecuteReader()

    End Using

Catch ex As Exception
    MsgBox(ex.ToString())
End Try
```

answered Jul 16 '16 at 14:02

fedeteka
170   2   11