

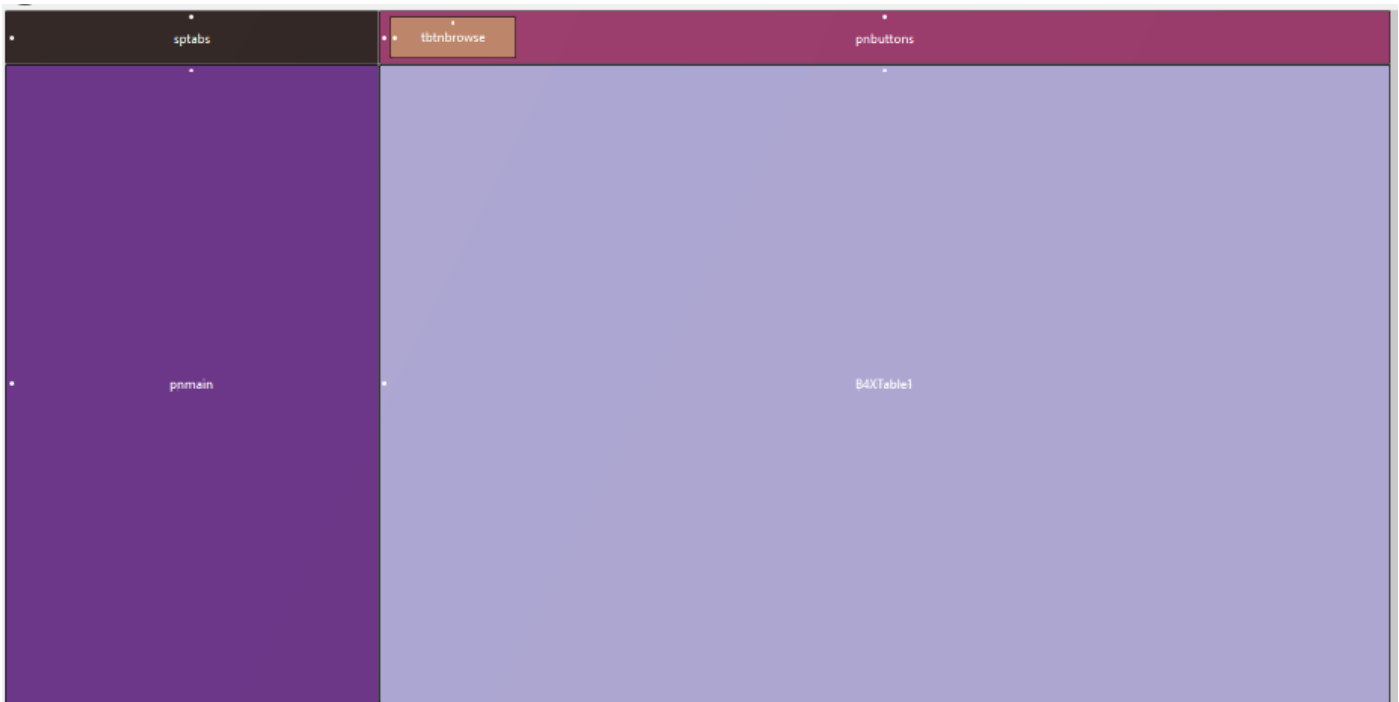
Report writer steps.

In this tutorial you can learn how to create a simple report writer application step-by-step.

Step 1 : transform the DBField_selector application into a database browser

Changes to the MainPage layout:

- Variants change: add a new variant 1300 by 650 and remove the old one (350 by 650).
- Remove the anchors from the sptabs, spmain and pnmain views and set their width to 350.
- Add a pane next to the sptabs view, name it **pnbuttons** and set the width at 950 and the height at 50.
- Add a togglebutton into the pnbuttons pane, name it **tbtnbrowse** and set the width at 120 and the height at 40. Set the text to "Browse data".
- Add a pane under the pnbuttons pane, name it **pnbrowse** and fill the rest of the layout (950 by 600).
- Add a B4Xtable into the pnbrowse pane, name it **B4Xtable1** and give it the same size as the pnbrowse. Don't forget to add the B4Xtable library to the Library manager panel in the IDE.
- Generate the pnbuttons, pnbrowse, tbtnbrowse and B4Xtable1 members. Select the SelectedChange event from the tbtnbrowse togglebutton.



Changes in the Main page:

- In the region Project Attributes set the #MainFormWidth to 1300.

#Region Project Attributes

#MainFormWidth: 1300

#MainFormHeight: 650

#AdditionalJar: sqlite-jdbc-3.7.2

#AdditionalJar: mysql-connector-java-5.1.27-bin.jar

#End Region

Changes in the B4XMainPage page:

- Make the pnbrowsers, pnbrowse, tbtnbrowse and B4Xtable1 variables Public (this will remove the warnings)

```
Public pnbrowse As Pane
```

```
Public pnbrowsers As Pane
```

```
Public tbtnbrowse As ToggleButton
```

```
Public B4XTable1 As B4XTable
```

- Add code to the tbtnbrowse_SelectedChange subroutine:
 - A try catch structure to get the errors logged.
 - An if test for the Selected state of the togglebutton.
 - An if test for the current tab in use: currtab = 0 for SQLite or currtab = 1 for MySQL
 - An if test on the contents of the tasqlitefile.text: if empty give a message and return
 - An if test on the contents of the tamysqldb.text: if empty give a message and return
 - An if test on the current tab = 0 to call the fill_B4XTable1_from_sqlite subroutine
 - An if test on the current tab = 1 to call the fill_B4XTable1_from_mysql subroutine

```
Private Sub tbtnbrowse_SelectedChange(Selected As Boolean)
```

```
Try
```

```
    If Selected Then
```

```
        If currtab = 0 Then
```

```
            If tasqlitefile.Text = "" Then
```

```
                xui.MsgboxAsync("Open a SQLite database file first.", "Browse data")
```

```
                Return
```

```
            End If
```

```
        End If
```

```
        If currtab = 1 Then
```

```
            If tamysqldb.text = "" Then
```

```
                xui.MsgboxAsync("Connect to the MySQL server first.", "Browse data")
```

```
                Return
```

```
            End If
```

```
        End If
```

```
        If currtab = 0 Then
```

```
            fill_B4XTable1_from_sqlite(tapath.text, tasqlitefile.text, tatable.text)
```

```
        End If
```

```
        If currtab = 1 Then
```

```
            fill_B4XTable1_from_mysql(tamysqldb.text, tamysqltable.text)
```

```
        End If
```

```
    End If
```

```
Catch
```

```
    Log(LastException)
```

```
End Try
```

```
End Sub
```

- Add the private subs fill_B4XTable1_from_sqlite and fill_B4XTable1_from_mysql.
- Add code to the fill_B4XTable1_from_sqlite subroutine:
 - Provide the arguments for the subroutine:
(fpath As String, fname As String, tname As String).
 - Clear the B4Xtable1 variable.
 - Loop over the fields list (clvfields0) and add a column for each field. The field types are text.
 - Initialize the data list (data) and the fldtypes list (fldtypes).
 - Run a dbsqlite query to get the fldtypes from the table and add the type to the fldtypes list.
 - Run a dbsqlite query to get all the records from the table.
 - Loop over the resultset, initialize a record list (rec), loop over the fields (clvfields0) and add the content to the record list if the fieldtype is not a BLOB, is not a GRAPHIC, is not Null. Otherwise add the fieldtype text as the content to the record list.
 - Add the record list to the data list.
 - After the loop close the resultset.
 - Use the B4Xtable1 method SetData(data) to add the data list to the B4Xtable1.

```

Private Sub fill_B4XTable1_from_sqlite(fpath As String, fname As String, tname As String)
    B4XTable1.Clear
    For i = 0 To clvfields0.Size - 1
        B4XTable1.AddColumn(clvfields0.GetValue(i), B4XTable1.COLUMN_TYPE_TEXT)
    Next
    Private data As List
    data.Initialize
    Dim fldtypes As List
    fldtypes.Initialize
    Private rs As ResultSet = dbsqlite.ExecQuery("PRAGMA table_info(" & tname & ")")
    Do While rs.NextRow
        Dim fld As String = rs.GetString("type").Trim
        fldtypes.Add(fld.Trim)
    Loop
    rs.Close
    Private rs1 As ResultSet = dbsqlite.ExecQuery("SELECT * FROM " & tname)
    Do While rs1.NextRow
        Dim rec As List
        rec.Initialize
        For i = 0 To clvfields0.Size - 1
            If fldtypes.Get(i) <> "BLOB" And fldtypes.Get(i) <> "GRAPHIC" Then
                If rs1.GetString2(i) <> Null Then
                    rec.Add(rs1.GetString2(i))
                Else
                    rec.Add("NULL")
                End If
            Else
                rec.Add(fldtypes.Get(i))
            End If
        Next
        data.Add(rec)
    Loop

```

```
rs1.Close
B4XTable1.SetData(data)
```

End Sub

You could run the application at this point to test it with a SQLite database. Open a SQLite database and click on the tbtnbrowse togglebutton ("Browse data").

- Add code to the fill_B4XTable1_from_mysql subroutine:
 - Provide the arguments for the subroutine:
(dbname As String,tname As String)
 - Clear the B4Xtable1 variable.
 - Loop over the fields list (clvmysqlfields) and add a column for each field. The field types are text.
 - Initialize the data list (data) and the fieldtypes list (fldtypes).
 - Run the dbmysql query to use the database.
 - Run a dbmysql query to get the fieldtypes from the table and add the type to the fieldtypes list.
 - Run a dbmysql query to get all the records from the table.
 - Loop over the resultset, initialize a record list (rec), loop over the fields (clvmysqlfields) and add the content to the record list if the fieldtype is not a date, is not Null. Otherwise if the fieldtype is a blob then set the content to "BLOB" and if it's not a blob then set the content to "NULL".
If the type is a date and the content is empty then MySQL can convert that field to "null". You have to change the initialization string from the dbmysql object in the get_schemas subroutine to the following:

```
dbmysql.Initialize2("com.mysql.jdbc.Driver",
"jdbc:mysql://localhost?characterEncoding=utf8&zeroDateTimeBehavior=convert
ToNull","root","root")
```

The first type = "date" test automatically turns the content to "null" and since the content is not a blob in the last test then the content will be "NULL".
 - Add the record list to the data list.
 - After the loop close the resultset.
 - Use the B4Xtable1 method SetData(data) to add the data list to the B4Xtable1.

```
Private Sub fill_B4XTable1_from_mysql(dbname As String,tname As String)
    B4XTable1.Clear
    For i = 0 To clvmysqlfields.Size -1
        B4XTable1.AddColumn(clvmysqlfields.GetValue(i),B4XTable1.COLUMN_TYPE_TEXT)
    Next
    Private data As List
    data.Initialize
    dbmysql.ExecQuery("USE " & dbname)
    Dim fldtypes As List
    fldtypes.Initialize
    Dim rstypes As ResultSet = dbmysql.ExecQuery2("SELECT column_type FROM
INFORMATION_SCHEMA.COLUMNS WHERE table_name = ? AND TABLE_SCHEMA = ?",Array As
String(tname,dbname))
```

```

Do While rstypes.NextRow
    fldtypes.Add(rstypes.GetString2(0))
Loop
Dim rs As ResultSet = dbmysql.ExecQuery("SELECT * FROM " & tname)
Do While rs.NextRow
    Dim rec As List
    rec.Initialize
    For i = 0 To clvmysqlfields.Size - 1
        If fldtypes.Get(i) = "date" Then
            Dim fldvalue As String = rs.GetString2(i)
        End If
        If fldtypes.Get(i) <> "date" Then
            Dim fldvalue As String = rs.GetString2(i)
        End If
        If fldvalue <> Null Then
            rec.Add(fldvalue)
        Else
            If fldtypes.Get(i) = "blob" Then
                rec.Add("BLOB")
            Else
                rec.Add("NULL")
            End If
        End If
    Next
    data.Add(rec)
Loop
rs.Close
B4XTable1.SetData(data)
End Sub

```

Now you can run the application again.

When you use a MySQL database you first need to connect to the database server (localhost). Select the database and table and click on the tbtnbrowse togglebutton ("Browse data").

The tbtnbrowse is a togglebutton so you can show or hide the pnbrowse pane. You need to change the code in the tbtnbrowse_SelectedChange subroutine. Before the second test for the currtab = 0 you can add:

```

pnbrowse.Visible = True

```

Before the last End If you can add an else statement and in it hide the pnbrowse pane:

```

Else

```

```

    pnbrowse.Visible = False

```

Run the application again and now you can show or hide the pnbrowse pane and the B4Xtable1 that's in it.

When you hide the browse pane you can use the space in the layout for other things...

More about that in the next step.