EMu Release Notes

ADO Reports

EMu 5.0

Document Version 1

EMu
Museum
Management
System

AXIELL
ARCHIVES LIBRARIES MUSEUMS

alm.axiell.com

Contents

SECTION 1

ADO Reports

Report generation and performance have been improved with EMu 5.0 and it is now possible to report directly to an Open Database Connectivity (ODBC) data source and to an ActiveX Data Objects (ADO) RecordSet object, bypassing the ODBC filtering process.

The new report options are:

- Crystal Reports: report directly in ODBC format, bypassing the ODBC filtering process.
- Crystal ADO: report using ADO RecordSets for Crystal (which are accessible via Crystal's ADO connector).
- Microsoft ADO: report using ADO RecordSets for Microsoft products.

Crystal and Microsoft reports (Excel, Power Point and Word) which currently connect to an ODBC data source can be modified to use an ADO RecordSet.
It remains possible to create reports by connecting directly to an ODBC data source.

Note

This document assumes familiarity with Report creation in EMu. Full details about Report Creation are available in the EMu Help: **Working with EMu records>Reports**.

SECTION 2

Crystal Reports

Creating a Crystal Report using the new ADO RecordSet is similar to creating a Crystal report with a direct ODBC connection. The main differences are in selecting the data source. This document describes the differences.

How to create a Crystal ADO Report

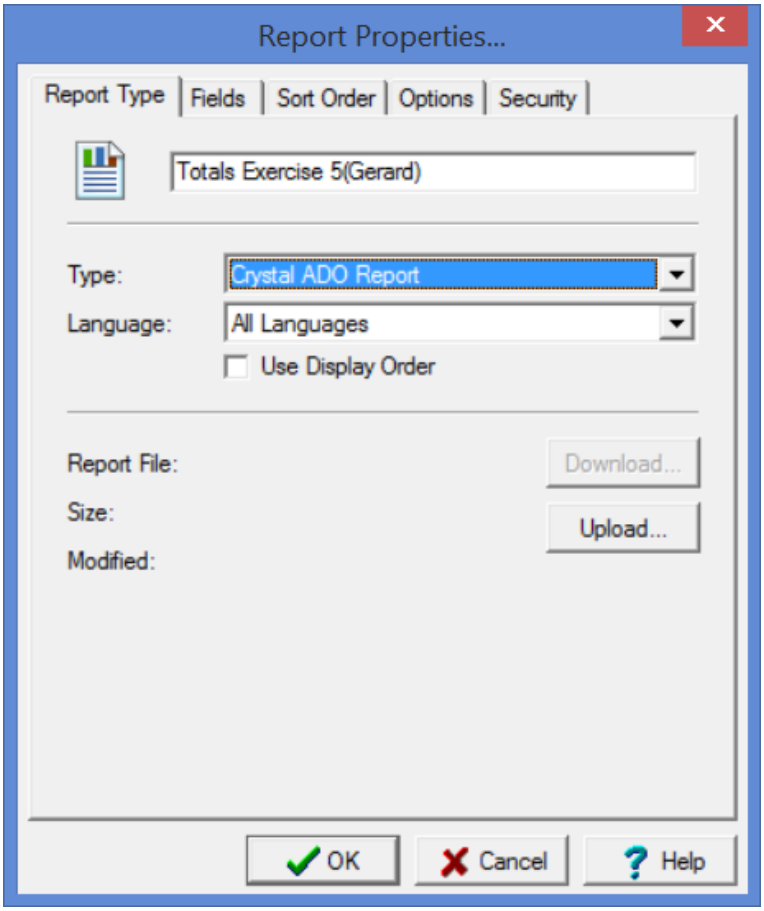
In EMu:

1. Search for or otherwise list a group of records on which to report.

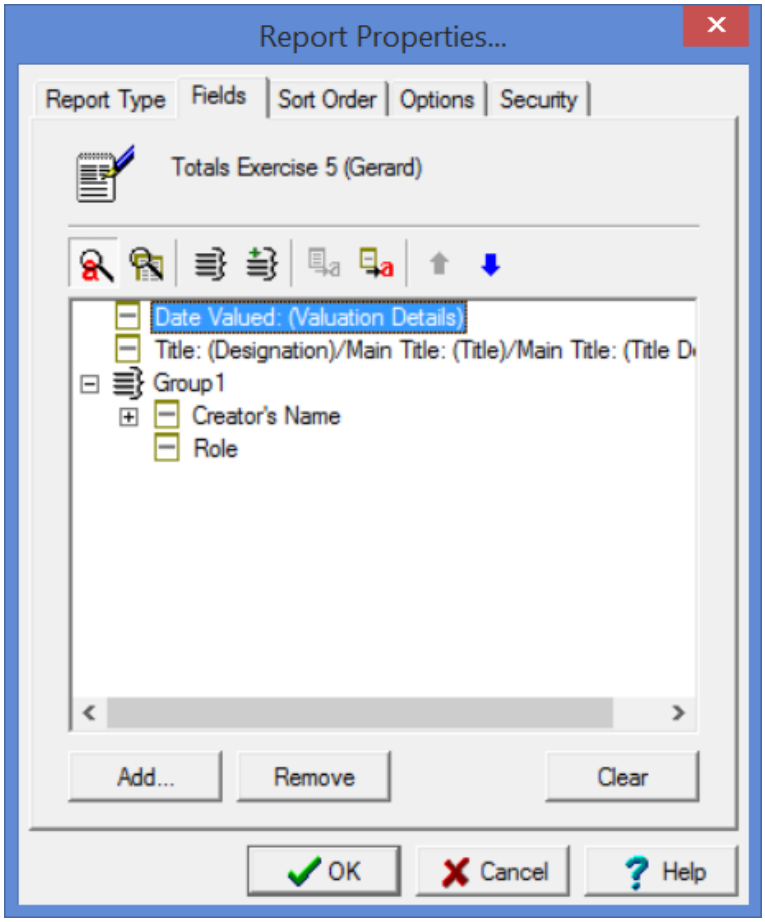
   When designing a Crystal ADO report the records in your initial record set must have a value in each field to be included in the report. If not, the field name will not appear in the list of available columns. Once the report is defined, it does not matter if a record does not have values in every field included in the report.

2. Click **Reports**  in the Tool bar to display the Reports box.
3. Click 

in the Reports box.
   The Report Properties box displays.
4. Enter a descriptive name for the Report in the top text field.
5. Select Crystal ADO Report from the *Type* drop list:

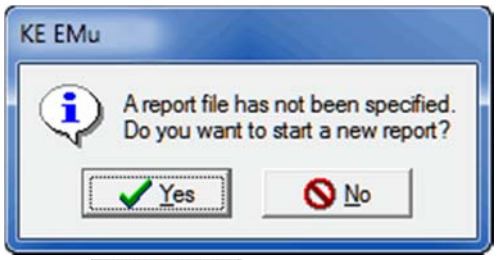


6. On the Fields tab, add the fields to be included in the report.
   In this example the fields selected are:



Note that a group was created using the **Create Group** button.

7. Make changes on the other tabs as required.
   See the EMu Help for details about setting a sort order, sort options, and security.
8. Click OK.
   The new report is added to the Reports box.
9. In the Reports box, select the new report and click Report All... to run the report for the first time.
   A message will display indicating that your report does not exist on the server. This is to be expected as the report has not yet been built in Crystal Reports:



10. Click Yes.

An xml file is generated and saved with the data from your record set. The location of this file can vary, but typically it can be found in:

`C:\Users\[`*your username*`]\AppData\Local\KESoftware\Reports\e[`*module name*`]`

For example, a report run in the Parties module, will save the xmldata file to:

`C:\Users\[`*your username*`]\AppData\Local\KESoftware\Reports\eparties`

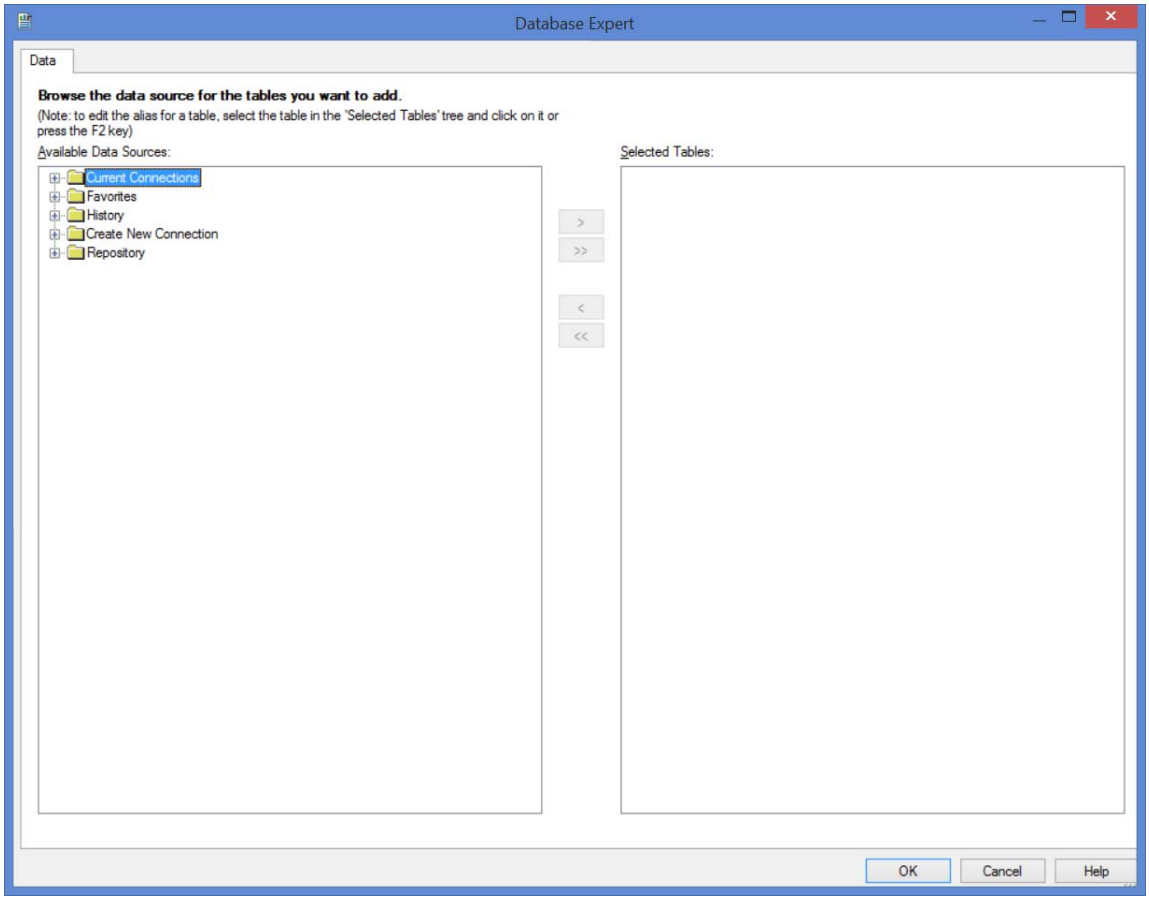
The Crystal Reports Designer application will open.

11. On the Start Page of the Crystal Reports Designer, select **Blank Report** under the New Reports heading

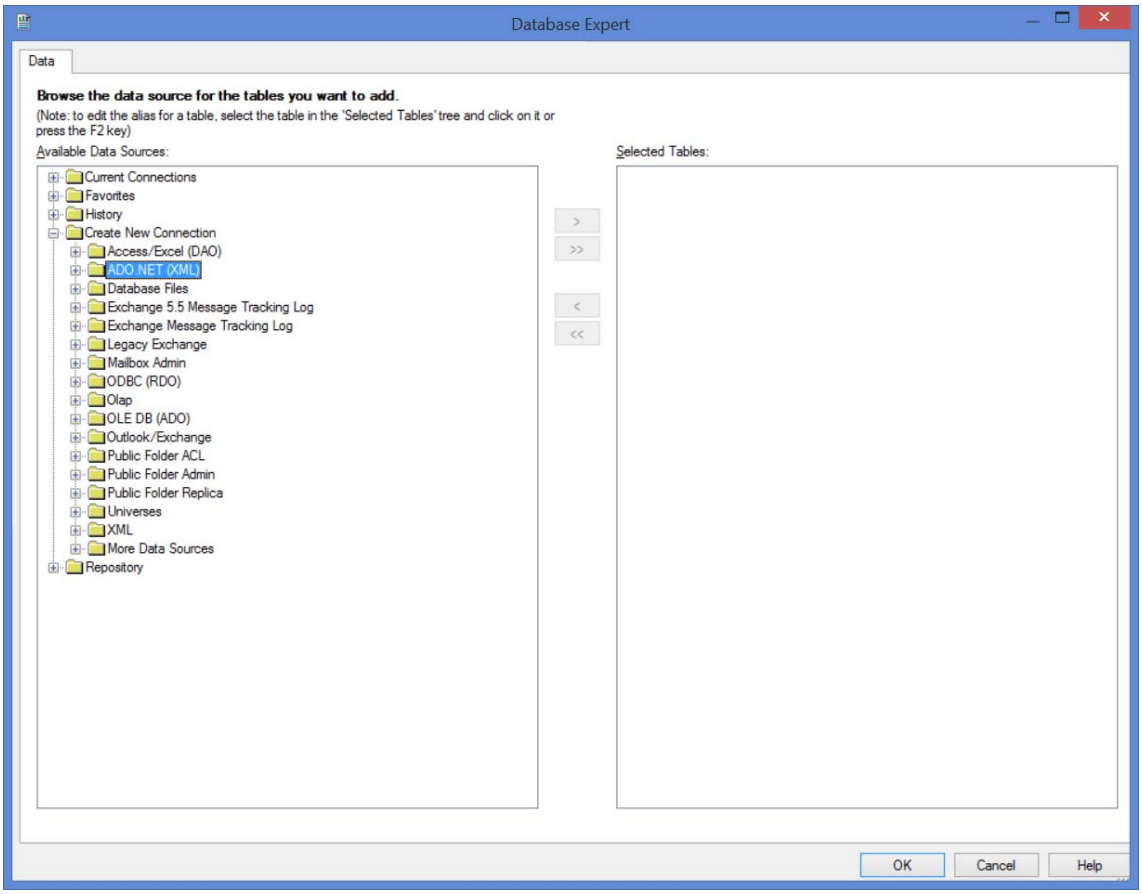
    -OR-

    Select **File>New>Blank Report** in the Menu bar.

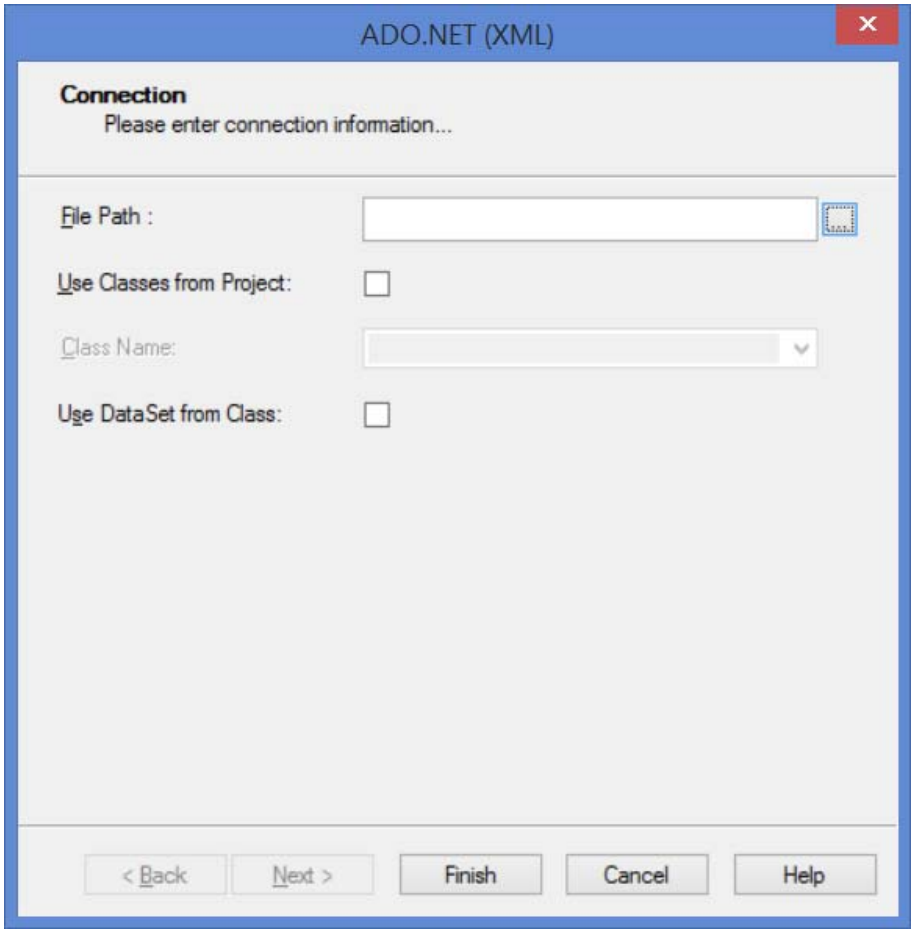
    The Database Expert box displays:



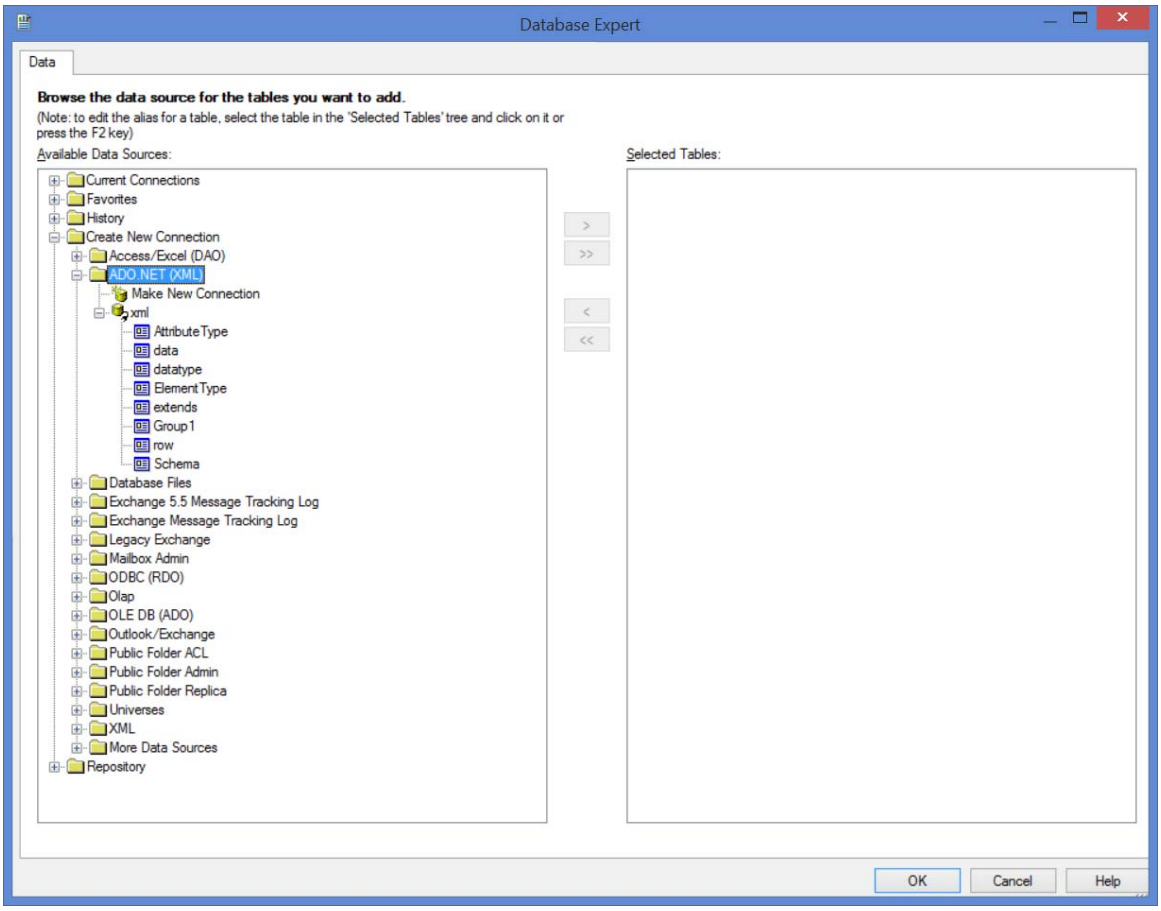
12. Double-click **Create New Connection** and click ⊞ beside **ADO.NET (XML)**:



The following screen will display:



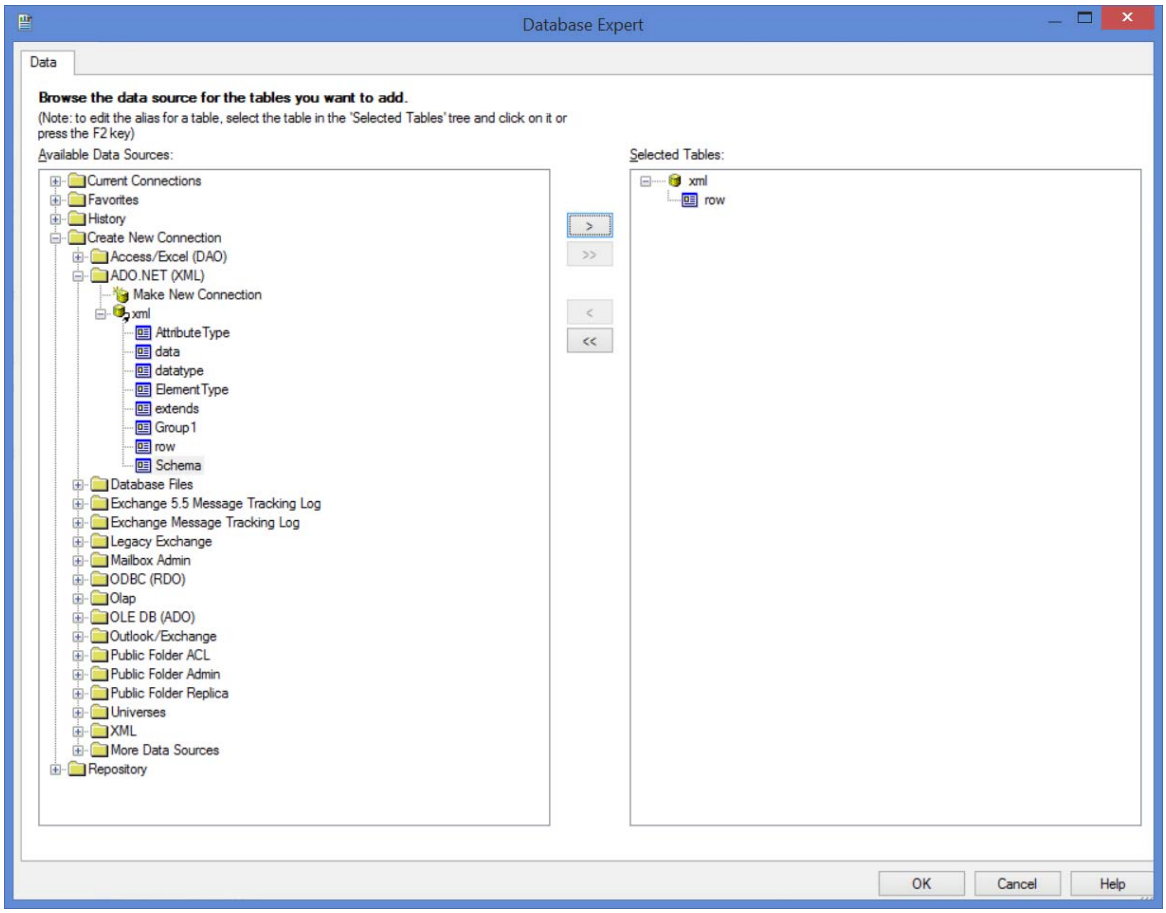
13. Click the button beside the *File Path* field to locate and select the `xmldata.xml` file created when the report was first run (Step 9).
    See Step 10 for details of the location of the `xmldata.xml` file.
14. Click  to return to the Database Expert:



`Group 1` contains values from fields that we grouped in the EMu report in this example (see Step 6). These fields are tables of values (they can hold more than one value). This data needs to be added to our report using a sub-report (see the EMu Help for details).

Field values from the Catalog are contained in the table called `row`.

15. Select **row** and add it to the *Selected Tables* pane:



16. Click OK.

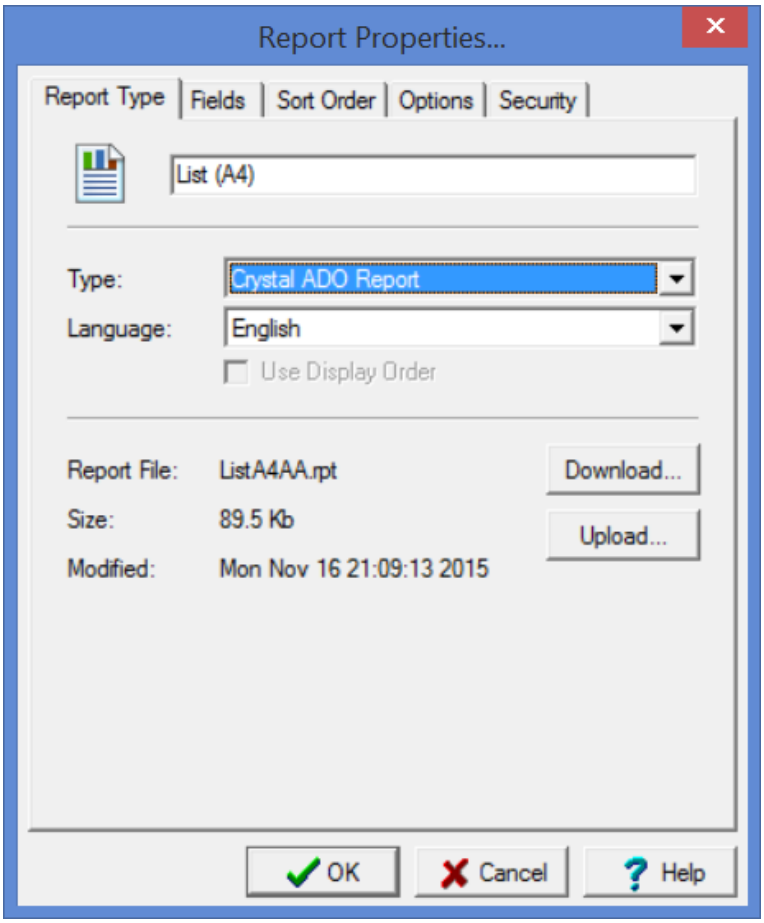
The Crystal Report Designer displays, ready for you to design your Crystal report. See the EMu Help for details of designing a Crystal Report.

It is important not to move the `xmldata.xml` file as this will cause problems when sharing the report with other users.

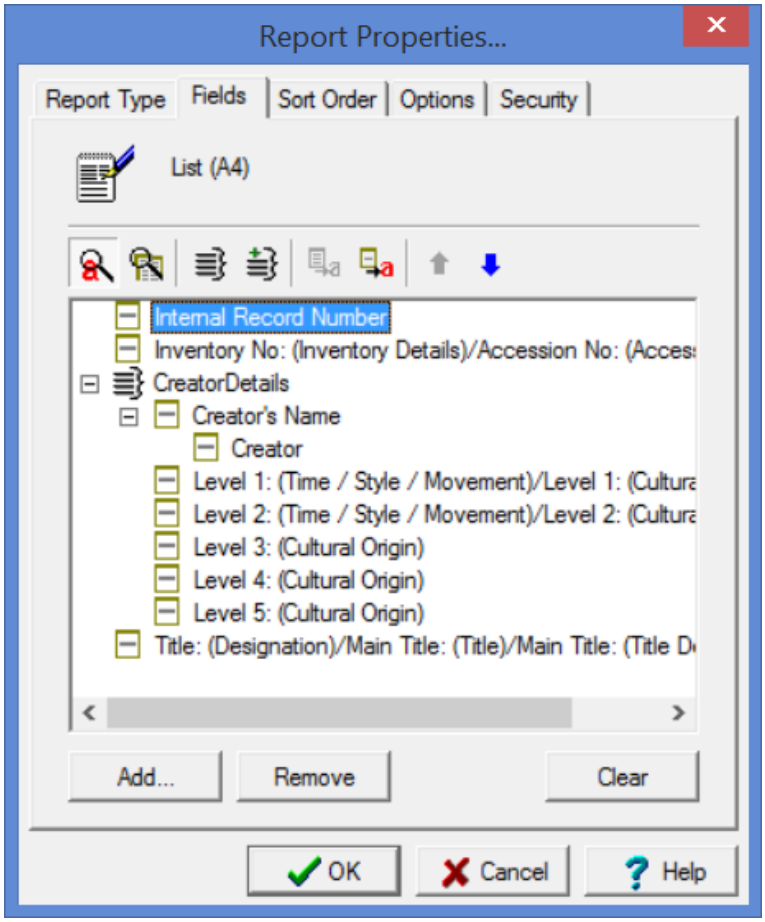
How to modify a Crystal Report to use ADO instead of ODBC

To modify a Crystal Report to use ADO rather than ODBC:

1. Open the Report Properties dialog for the report.
   This example uses the default report `List (A4)`.
2. Select **Crystal ADO Report** from the Type drop list:



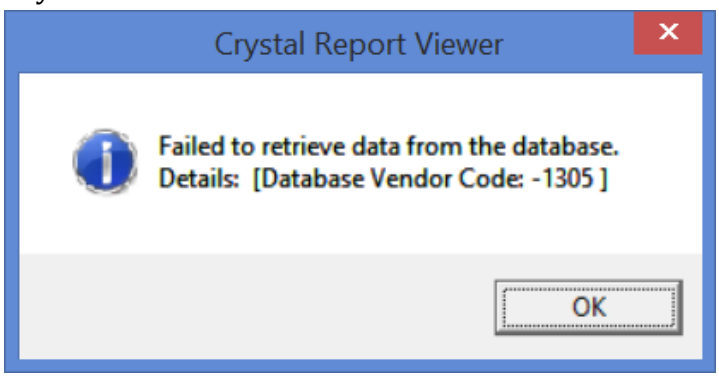
The fields for this report are:



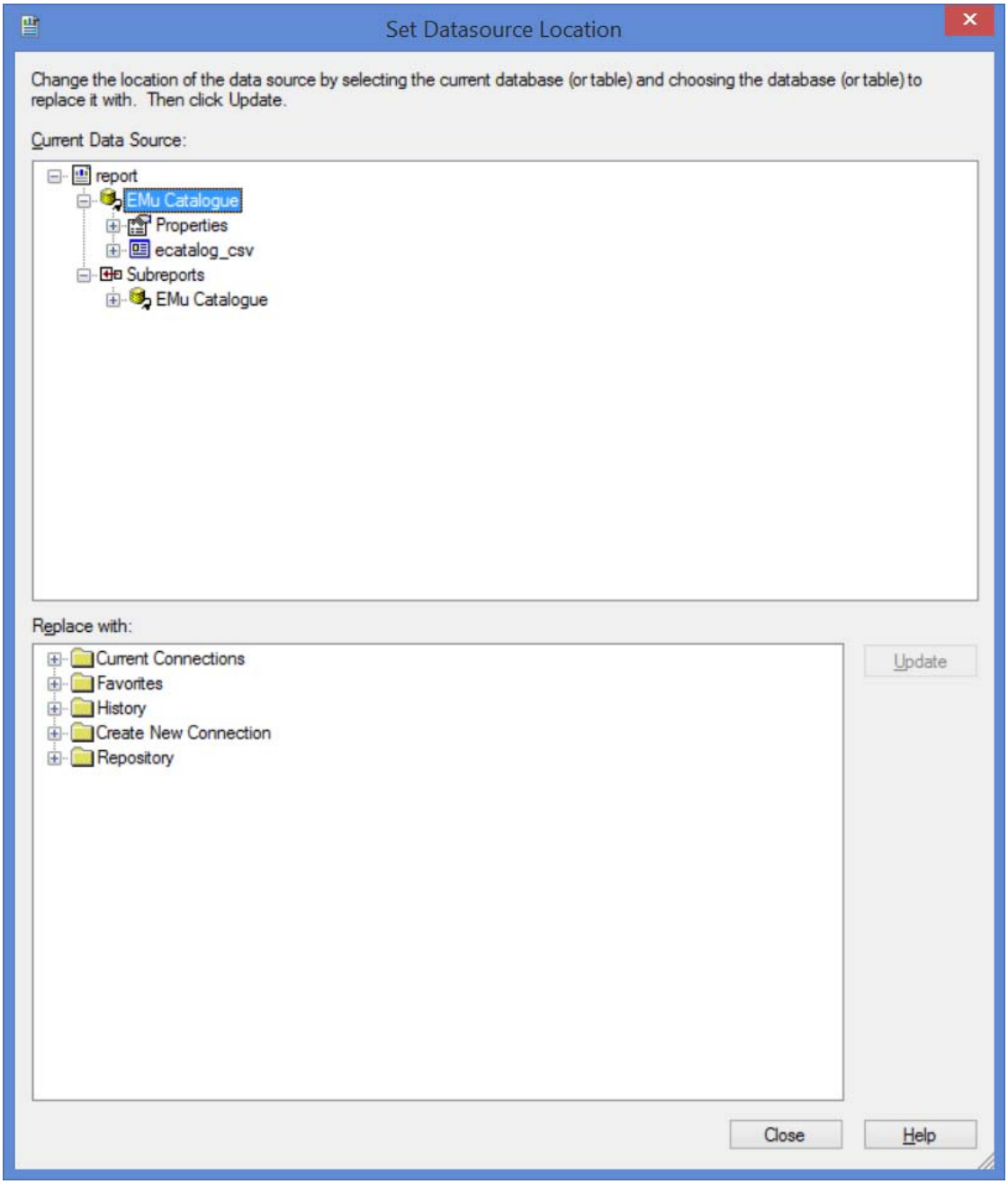
Two tables are generated in this report.

3. Click  and run the report.

Crystal will create the ADO record set and the following error will display:

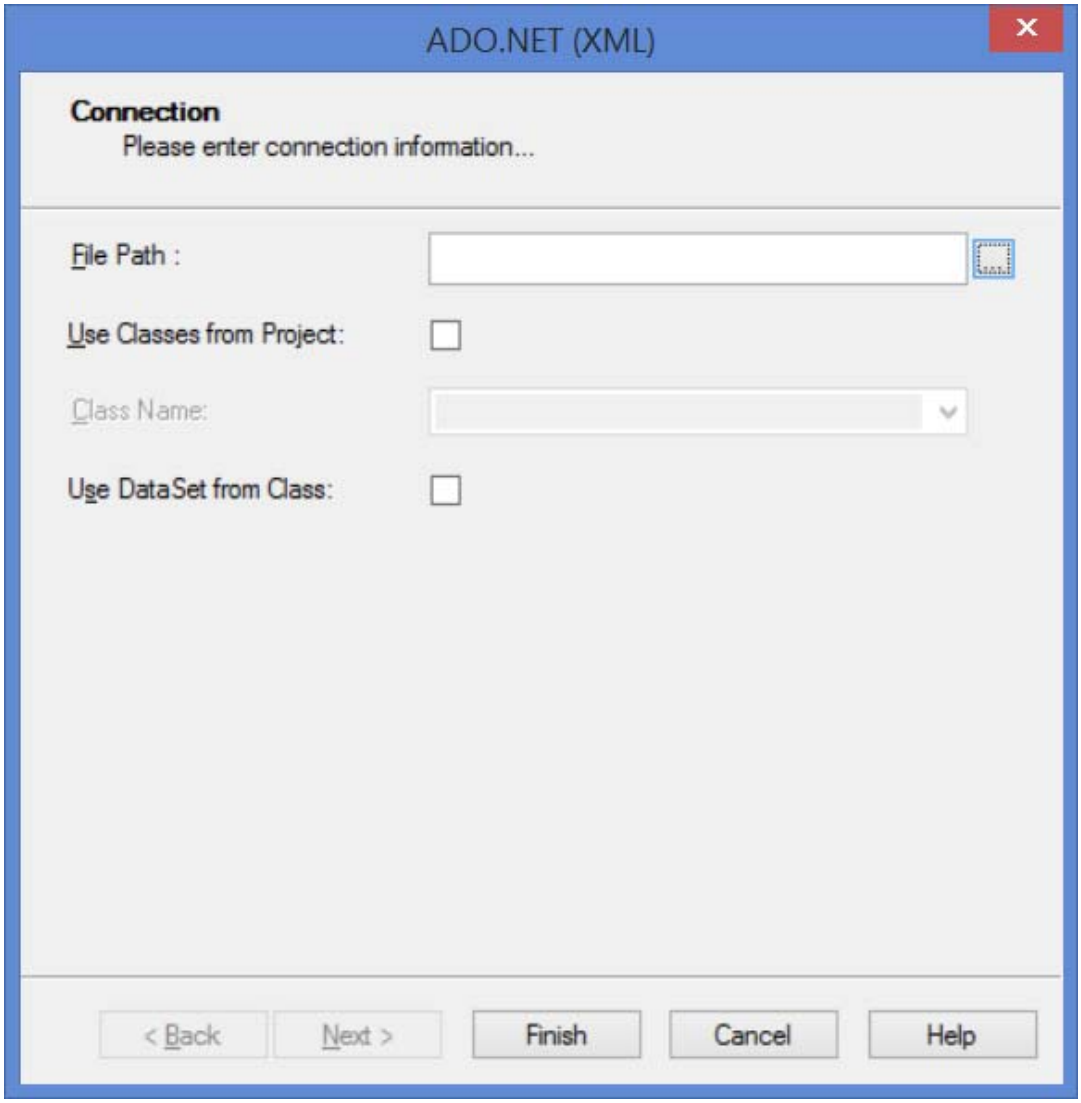


4. Open the Crystal report in the Crystal Report Designer and select the **Database>Set Datasource Location** menu option.
The Set Datasource Location dialog will display:

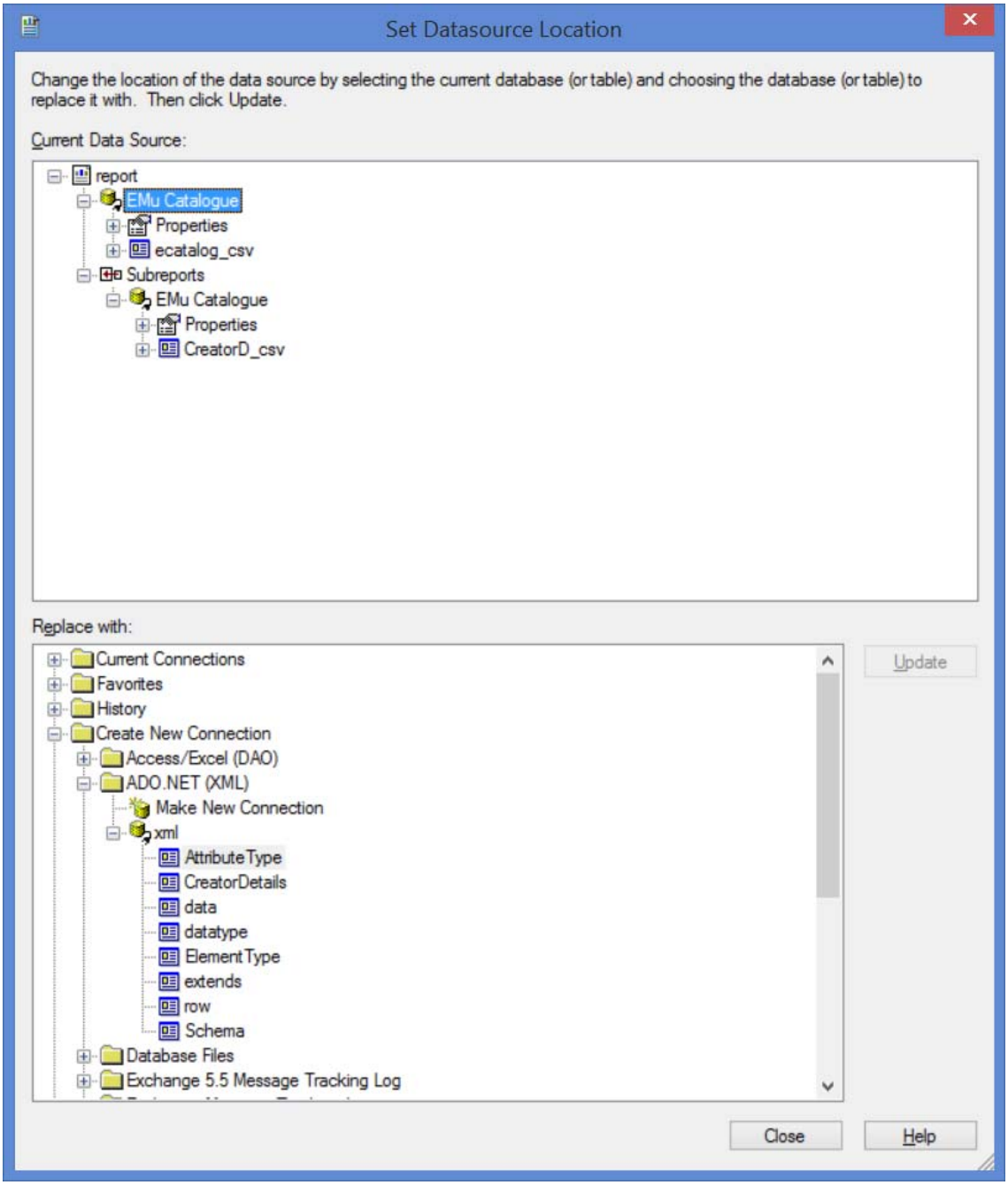


5. Select **Create New Connection** in the *Replace with* pane and click ⊞ beside **ADO.NET (XML)**.

   The following screen will display:



6. Click the button beside the *File Path* field to locate and select the `xmldata.xml` file created when the report was run.
   The location of this file can vary, but typically it can be found in:
   `C:\Users\[`*your username*`]\AppData\Local\KESoftware\Reports\e[`*module name*`]`
   For example, a report run in the Parties module, will save the xmldata file to:
   `C:\Users\[`*your username*`]\AppData\Local\KESoftware\Reports\eparties`
7. Click Finish.
   You are returned to the Set Datasource Location dialog:



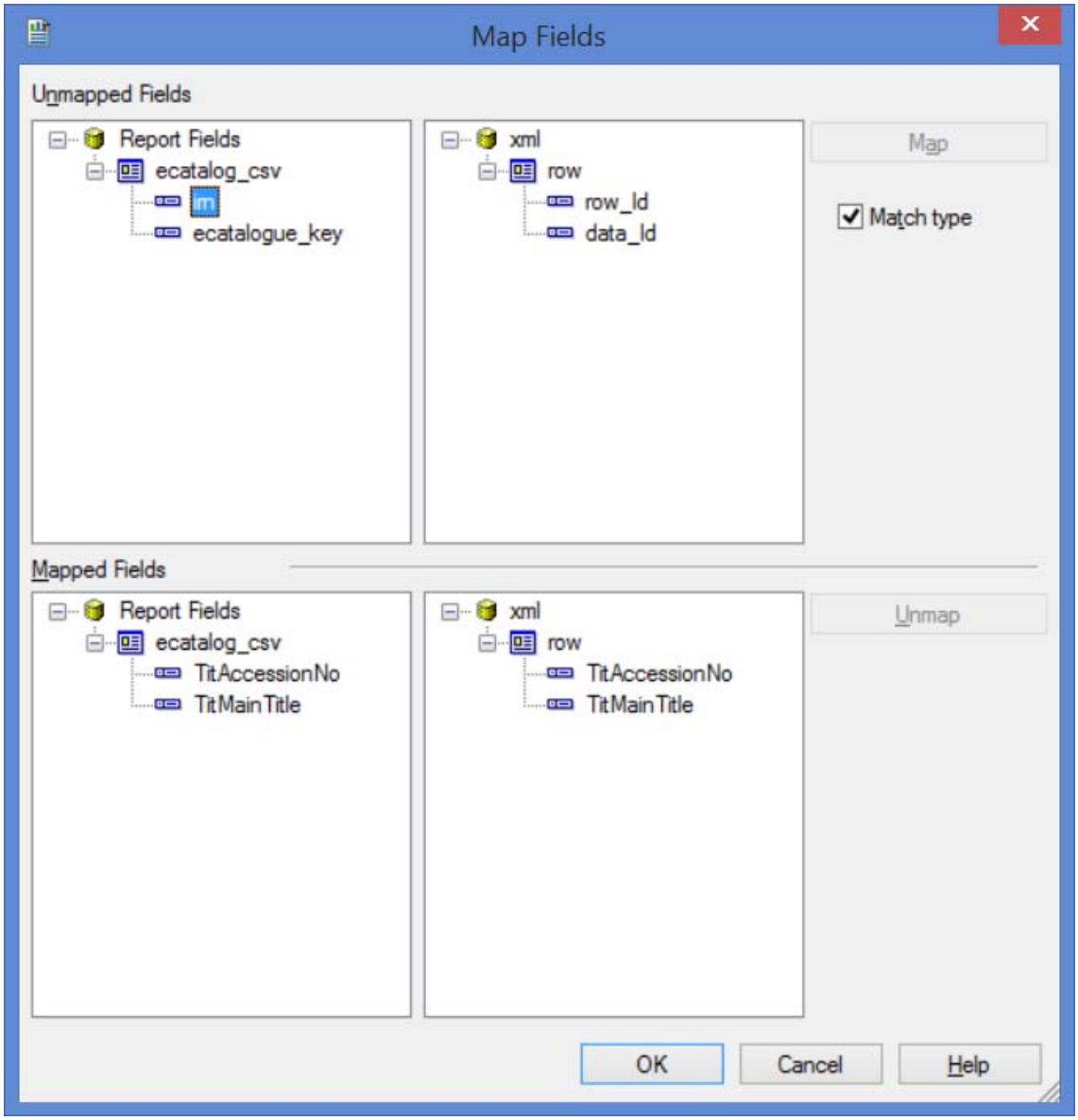
Next it is necessary to map fields from the old ODBC data source to the new ADO RecordSet.

In this example there are two tables to map and one sub-report.

8. To map the old ODBC Catalog fields to the new Catalog table, click **ecatalogue_csv** in the *Current Data Source* pane and then click the **row** table in the *Replace with* pane.

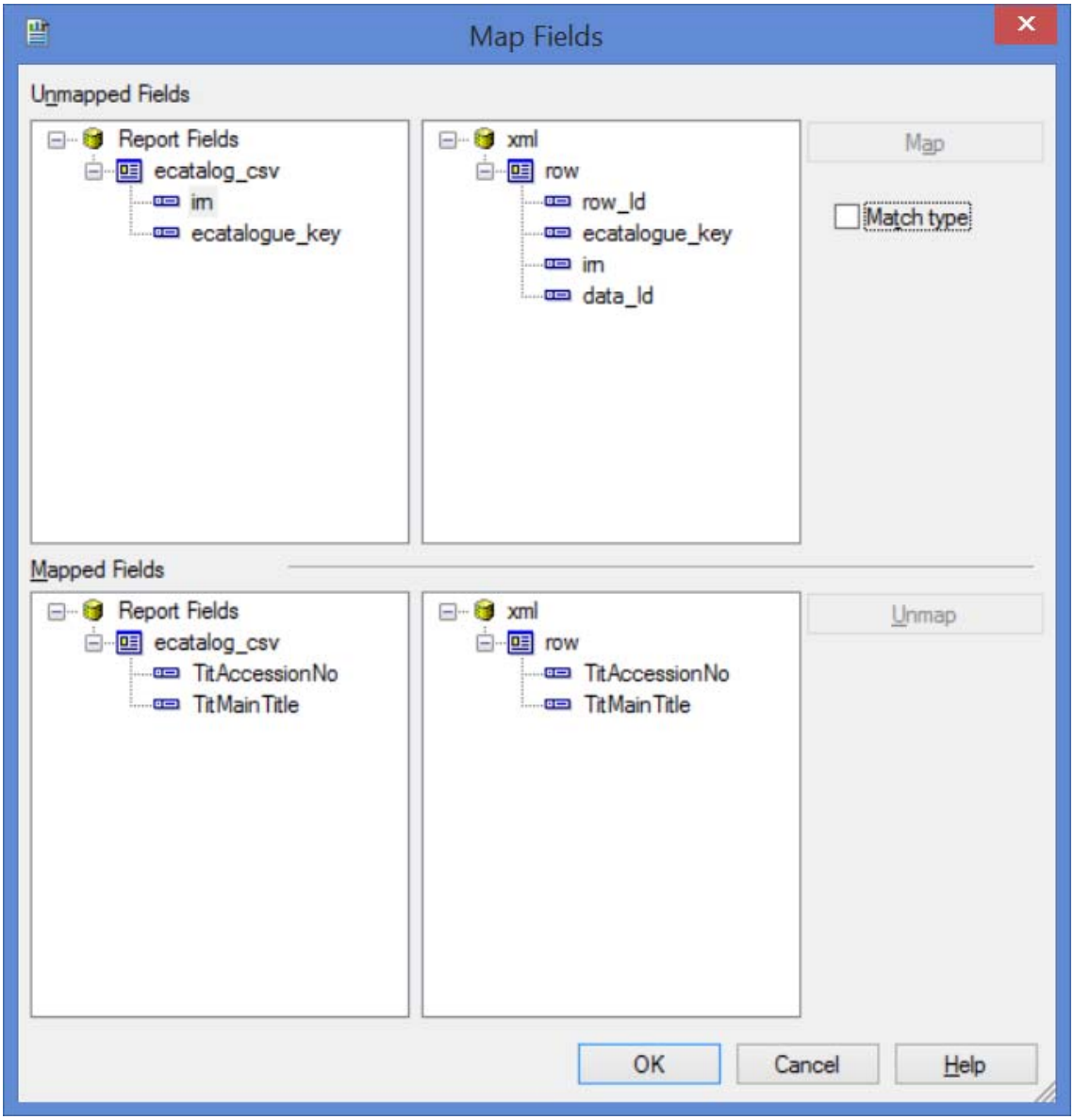
   The Update button will be enabled.

9. Click the **Update** button and the Map Fields dialog will display:



Fields with the same name will be mapped automatically.

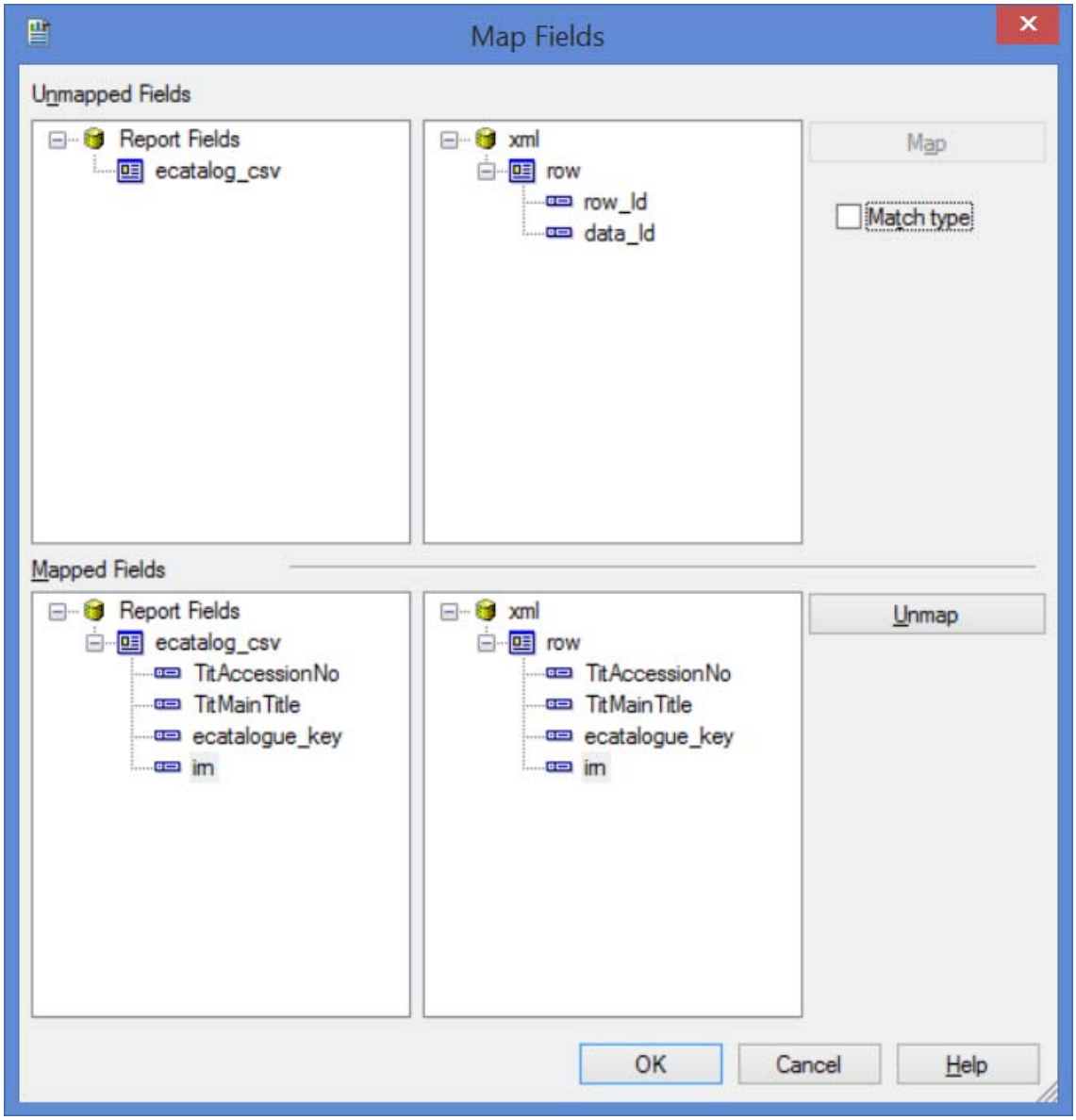
10. Uncheck the **Match type** check box to reveal more fields in the *Unmapped Fields* pane:



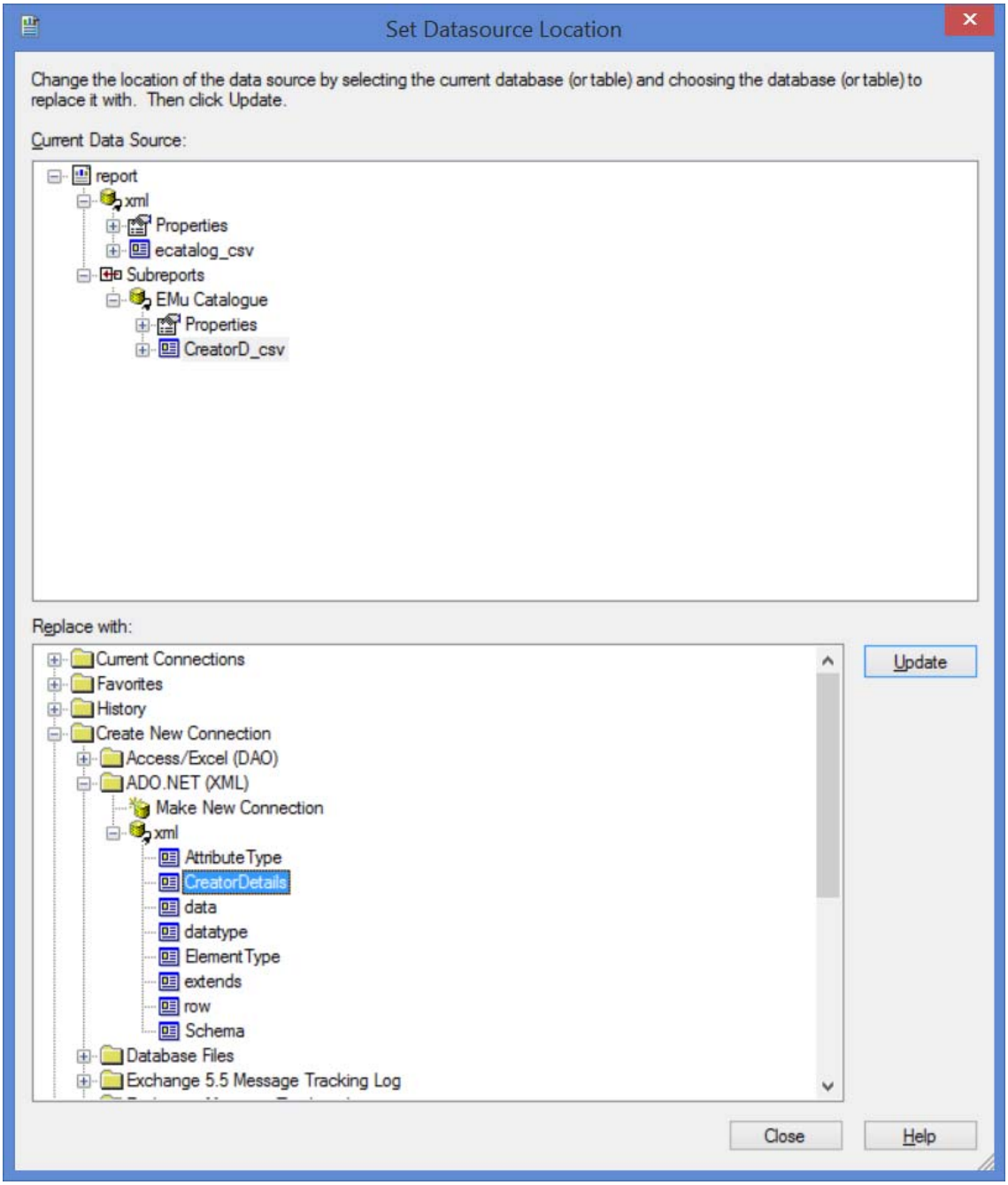
11. Complete mapping fields in the *Unmapped Fields* pane.

    In this example we map `ecatalogue_key` to `ecatalogue_key` and `irn` to `irn` by selecting both fields to map and clicking the **Map** button.

    Once mapped, fields will be moved to the *Mapped Fields* pane:



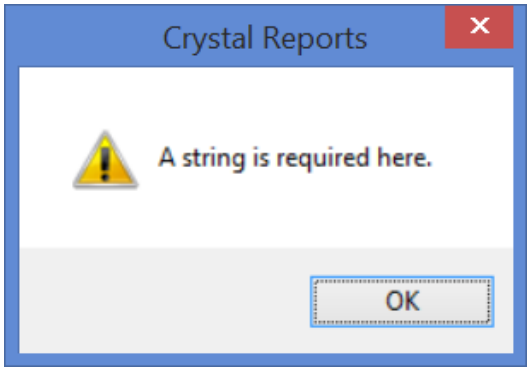
12. Click OK when all fields are mapped.
    You are returned to the Set Datasource Location dialog.
13. Repeat the mapping process for all fields (in this example, mapping fields in the `CreatorD_csv` table to the ADO table `CreatorDetails`):



14. Once all fields have been remapped in all tables click **Close**.

    You are returned to the Crystal design window.

    If you refresh report data at this stage and you have a sub-report object, you will probably receive an error regarding sub-report links, e.g.:



Click  to open the Record Selection Formula Editor. Change the link key field used by the old ODBC table to the link key field referenced by the ADO RecordSet:

The report should now work correctly.

SECTION 3

Microsoft Excel

The following examples demonstrate how to create a basic Excel report using VBA. Please note that it is not the intention of this document to teach VBA.
Excel 2013 was used to create these reports.

How to create an Excel Report using the ADO RecordSet

With ODBC data sources there is an option in Excel to open a connection without writing Visual Basic code. This is not the case when making a connection to an ADO record set and it is necessary to write VB code.

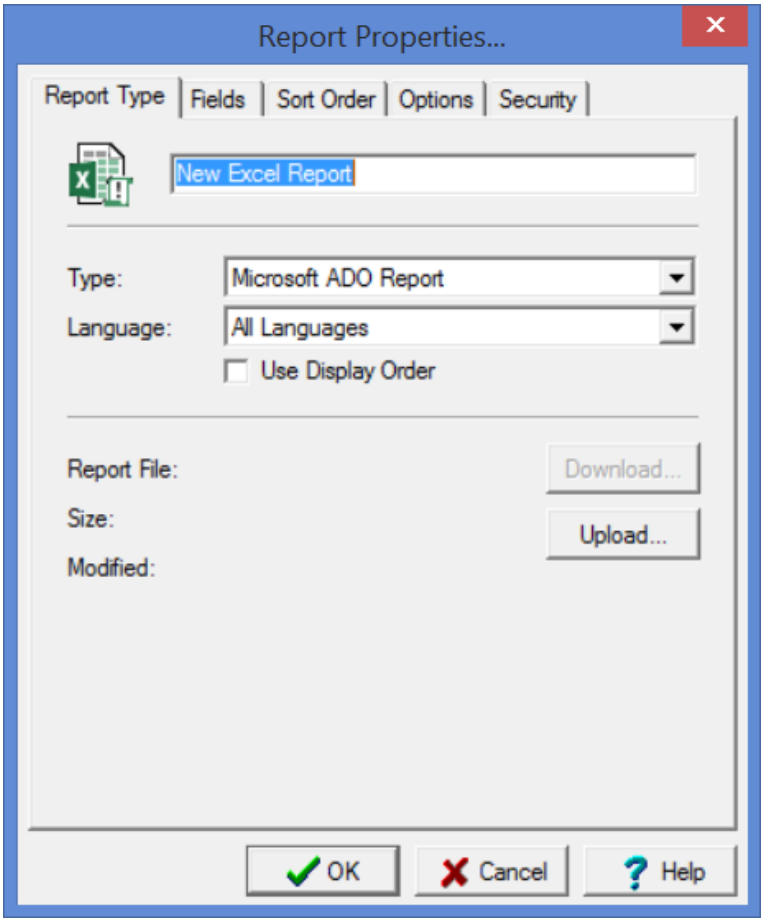
Step 1: Create a new report in EMu

This first example is a simple report on single value fields from the Catalog module. The VBA code provided in this example will automatically populate headings and row data for each column selected.

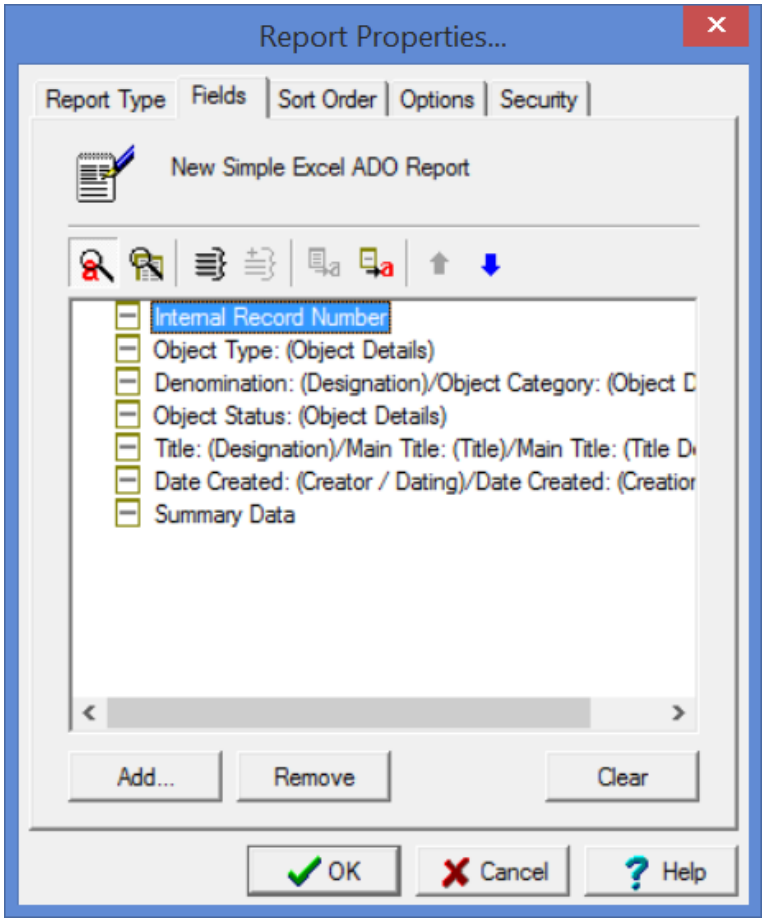
In EMu:

1. Search for or otherwise list a group of records on which to report.
2. Click **Reports**  in the Tool bar to display the Reports box.
3. Click 

in the Reports box.
The Report Properties box displays.
4. Enter a descriptive name for the Report in the top text field.
5. Select Microsoft ADO Report from the *Type* drop list:



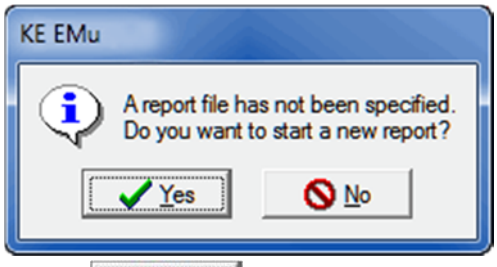
6. On the **Fields** tab, add the fields to be included in the report.
Fields selected in this example are:



7. Make changes on the other tabs as required.
   See the EMu Help for details about setting a sort order, sort options, and security.
8. Click 

   .
   The new report is added to the Reports dialog box.
9. Select the new report and click 

   to run the report for the first time.
   A message will display indicating that your report does not exist on the server. This is to be expected as the report has not yet been built in Excel:



10. Click 

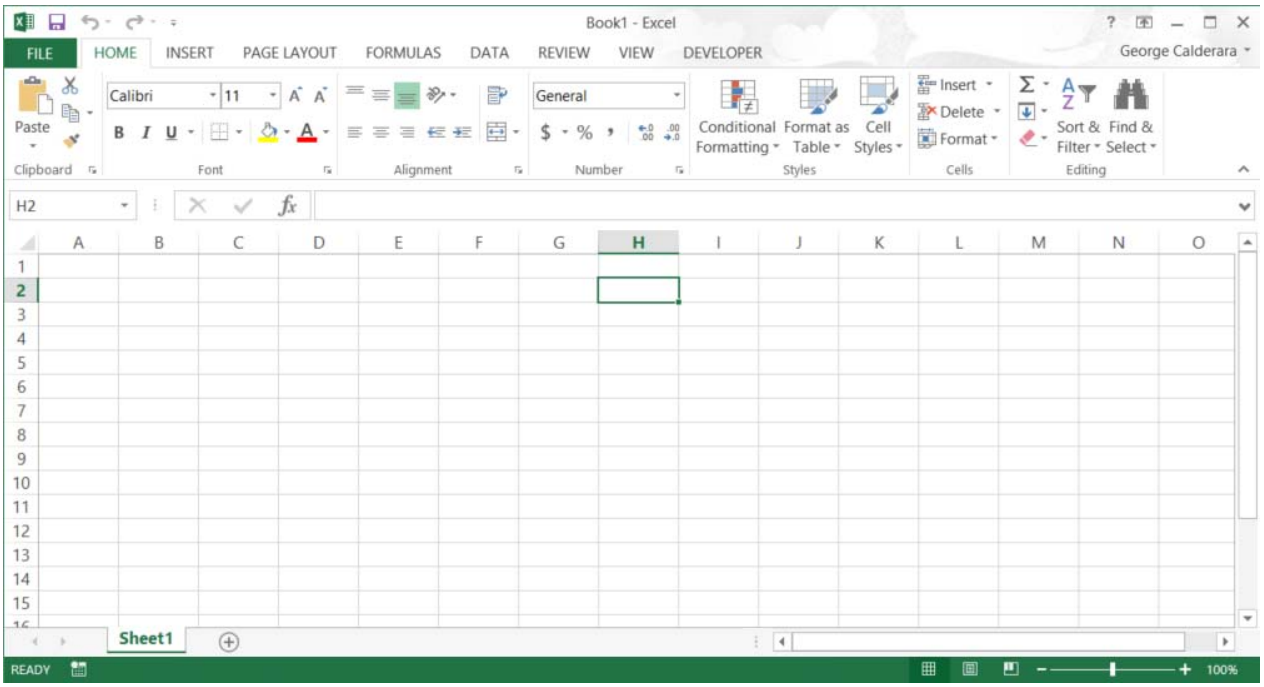
    .
    An xml file is generated and saved with the data from your record set. The location of this file can vary, but typically it can be found in:
    `C:\Users\[`*your username*`]\AppData\Local\KESoftware\Reports\e[`*module*

name]

For example, a report run in the Parties module, will save the xmldata file to:

`C:\Users\[`*your username*`]\AppData\Local\KESoftware\Reports\eparties`

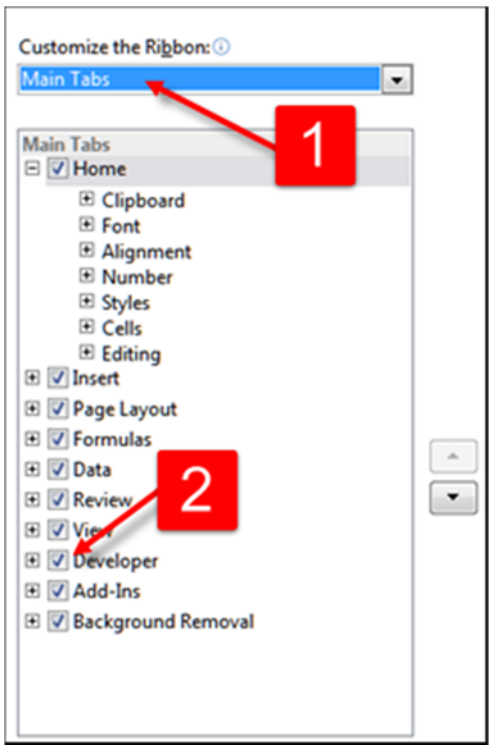
Microsoft Excel will open with a blank worksheet as follows:



Ensure that Excel is setup correctly

If the Developer tab does not display in the Ribbon:

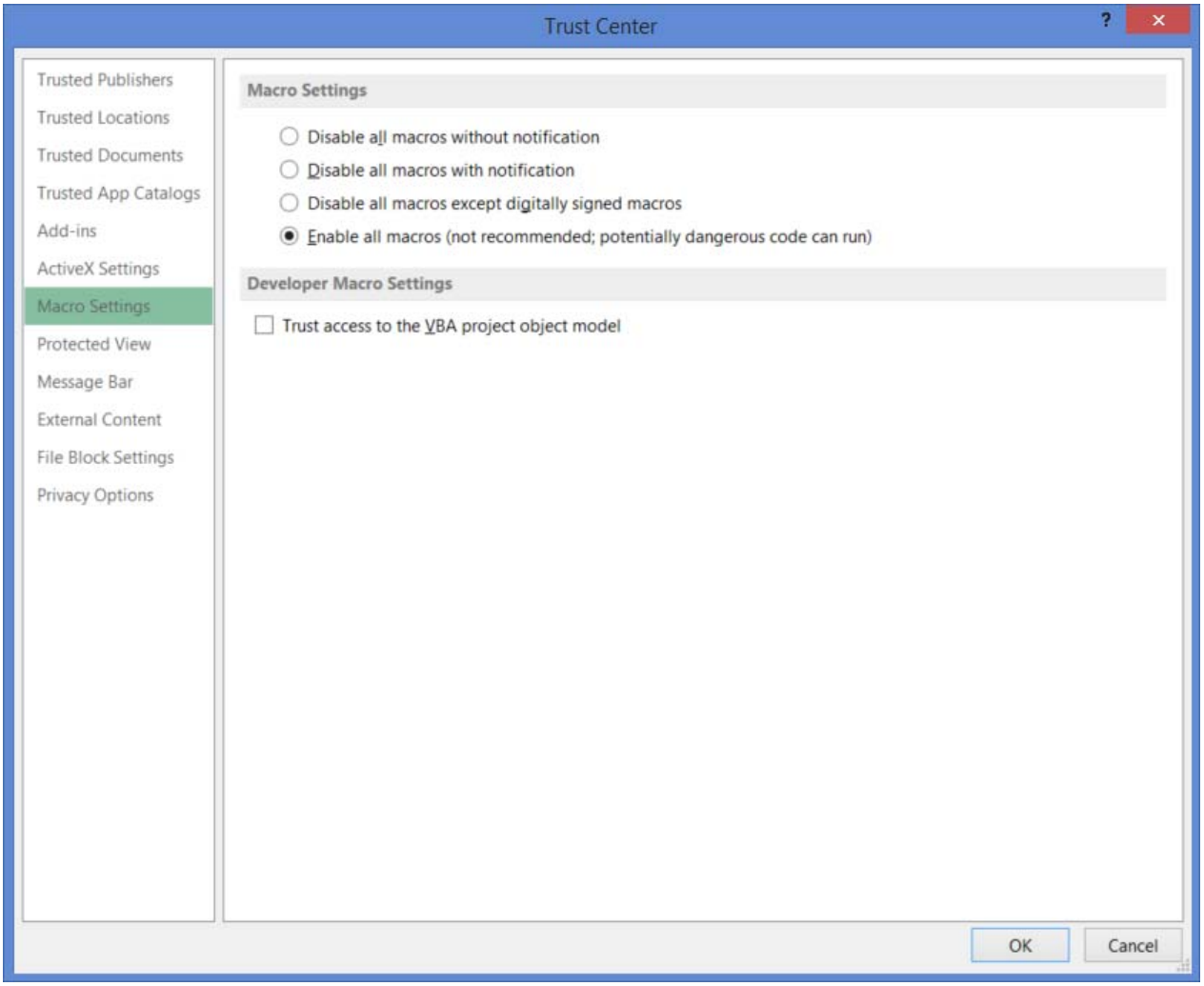
1. Click **File>Options>Customize Ribbon**.
2. With **Main Tabs** selected from the *Customize the Ribbon* drop list (1), select the **Developer** check box (2):



In order to run the macros that we will create with our reports, we need to ensure that the Security level in Excel is appropriate:

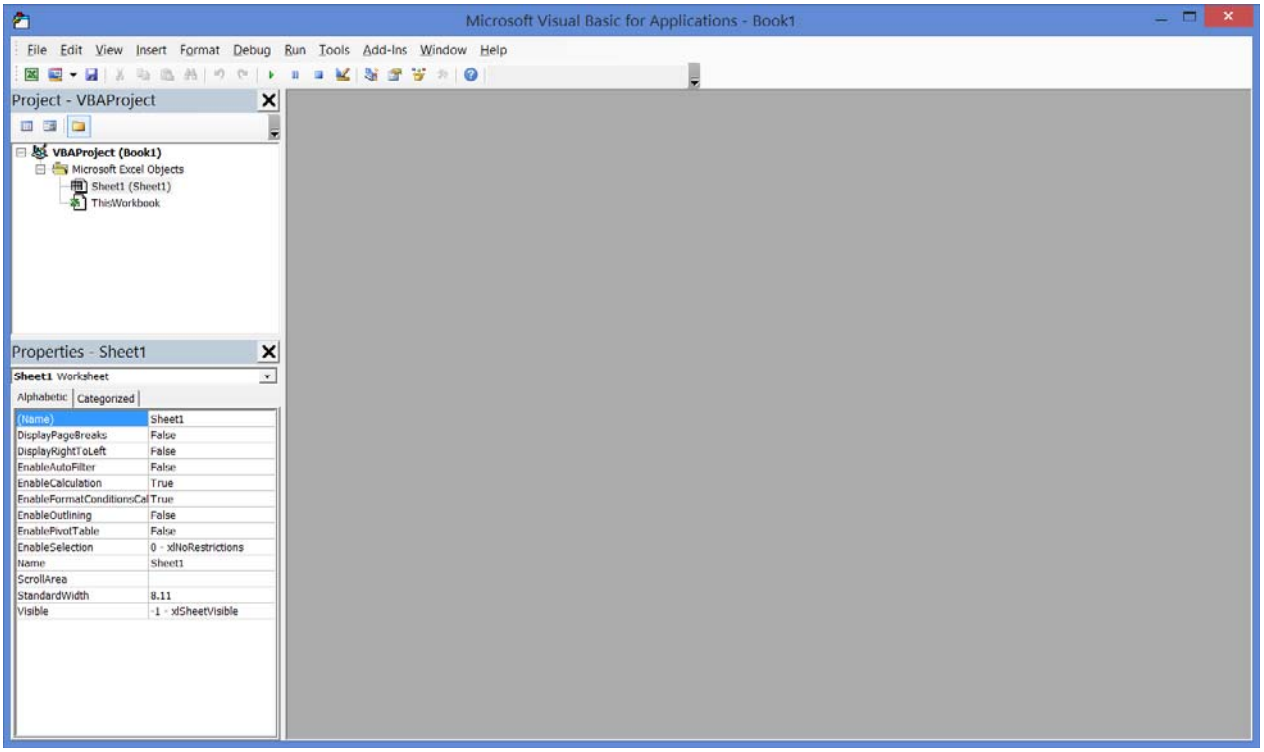
1. On the Developer tab, click 

.
2. Enable all macros:



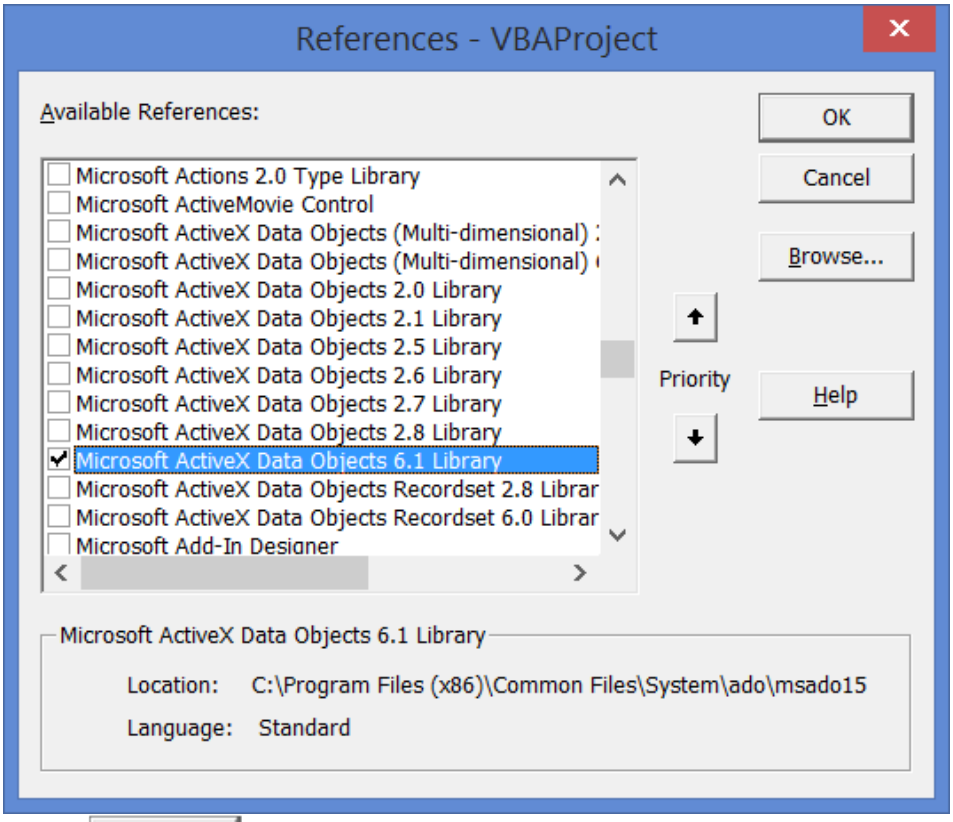
3. Click OK to close the Trust Center.

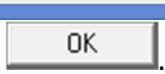
4. On the Developer tab, click Visual Basic.
   The following screen displays:



5. Ensure that the Microsoft ActiveX Data Objects Library is available:
   5.1. Select **Tools>References** in the Menu bar

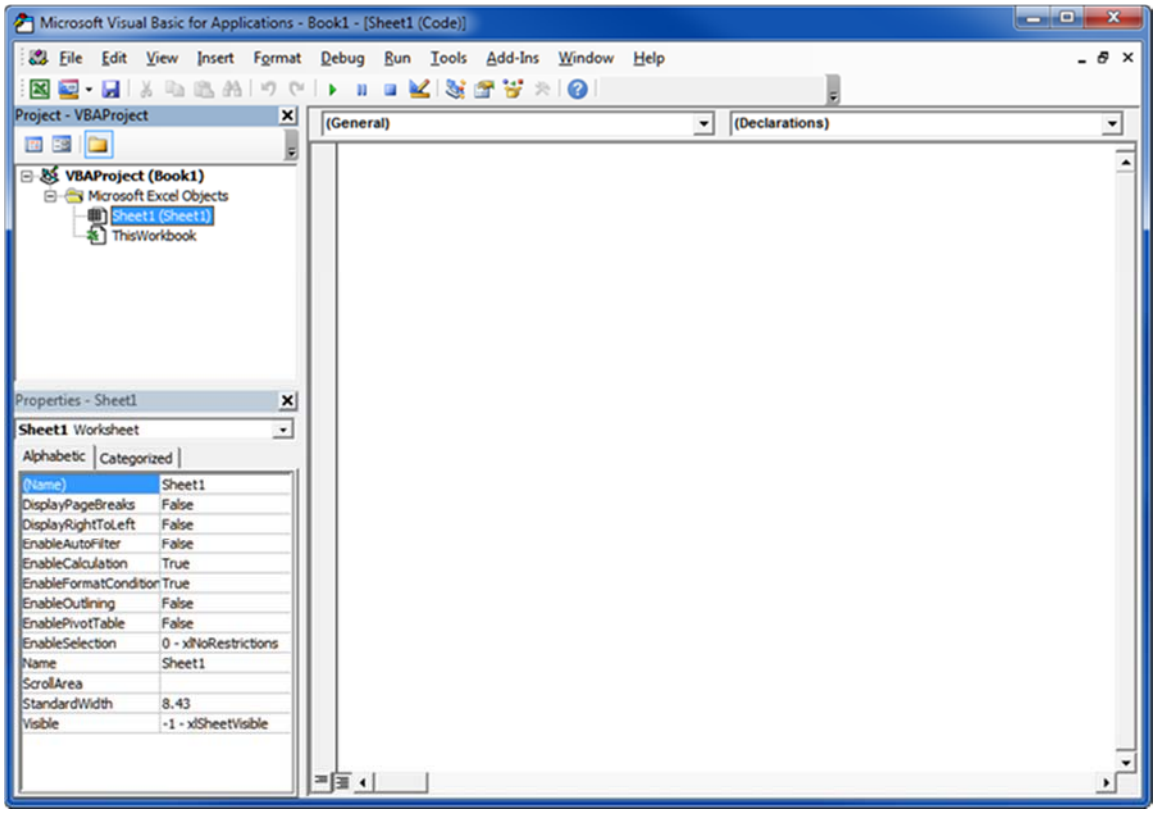
   In the References – VBAProject dialog that displays, make sure that the following checkbox is checked:


References - VBAProject
Available References:
OK
Cancel
Browse...
Priority
Help
Microsoft Actions 2.0 Type Library
Microsoft ActiveMovie Control
Microsoft ActiveX Data Objects (Multi-dimensional)
Microsoft ActiveX Data Objects (Multi-dimensional)
Microsoft ActiveX Data Objects 2.0 Library
Microsoft ActiveX Data Objects 2.1 Library
Microsoft ActiveX Data Objects 2.5 Library
Microsoft ActiveX Data Objects 2.6 Library
Microsoft ActiveX Data Objects 2.7 Library
Microsoft ActiveX Data Objects 2.8 Library
Microsoft ActiveX Data Objects 6.1 Library
Microsoft ActiveX Data Objects Recordset 2.8 Librar
Microsoft ActiveX Data Objects Recordset 6.0 Librar
Microsoft Add-In Designer
Microsoft ActiveX Data Objects 6.1 Library
Location: C:\Program Files (x86)\Common Files\System\ado\msado15
Language: Standard

5.2. Click 
OK
.

Step 2: Design the report in Excel

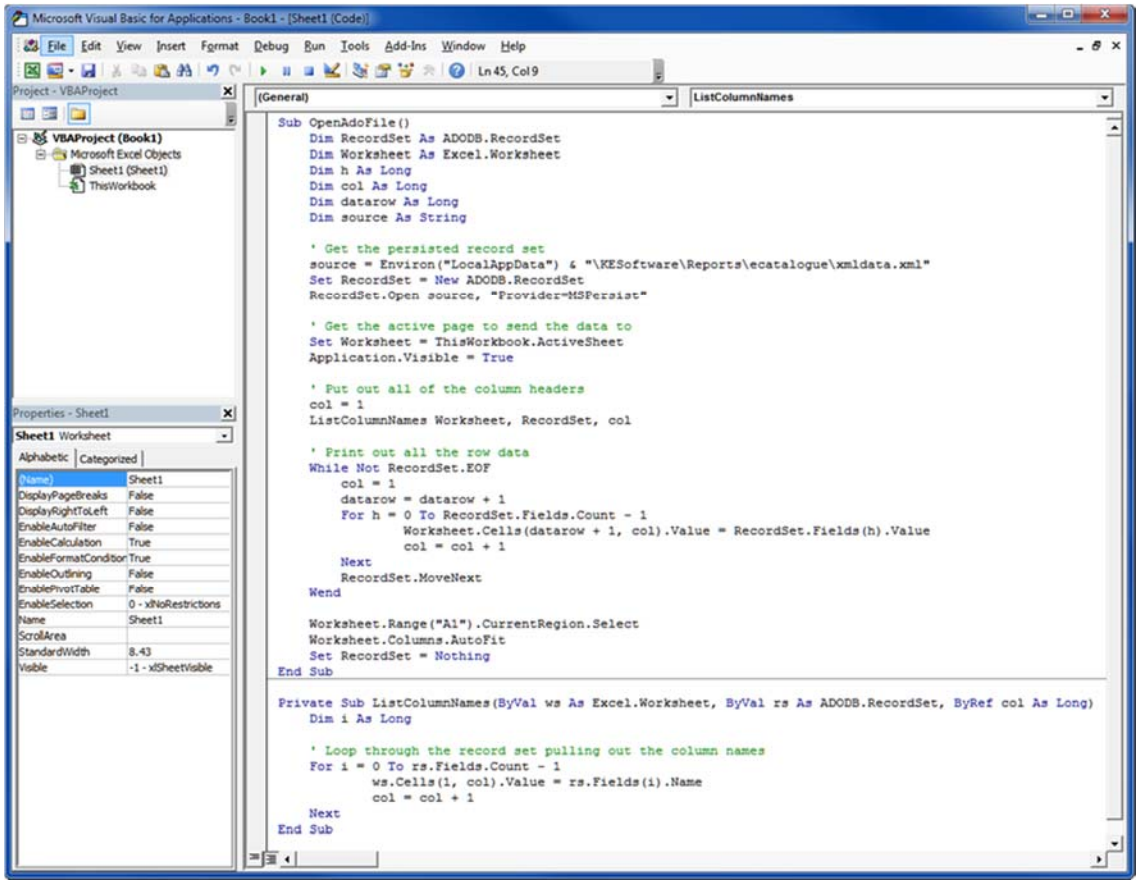
1. Double-click **Sheet1** in the VBAProject pane:



2. Copy and paste the following VB code:

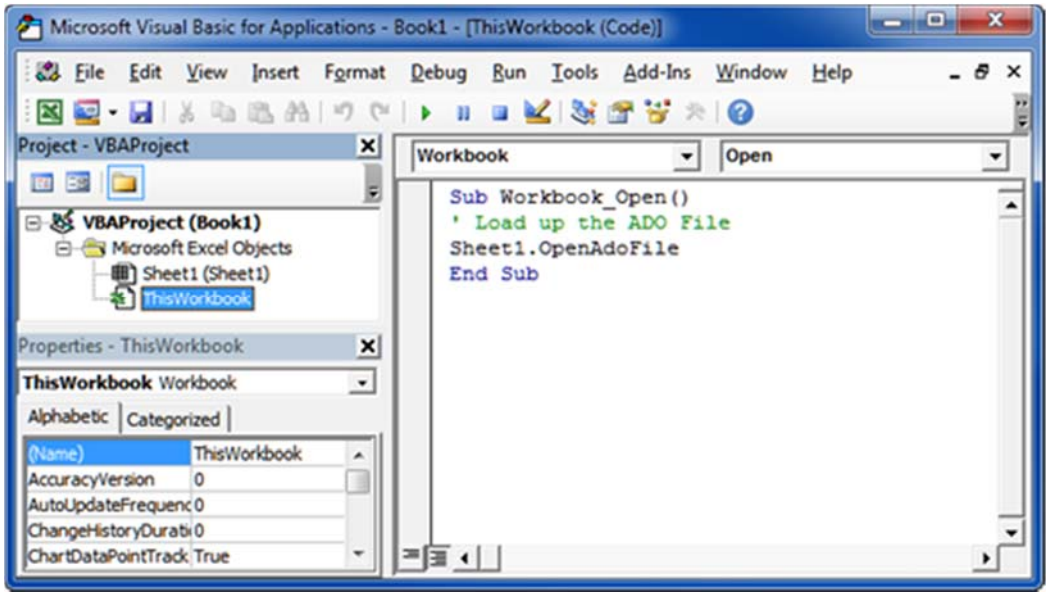
```
Sub OpenAdoFile()
    Dim RecordSet As ADODB.RecordSet
    Dim Worksheet As Excel.Worksheet
    Dim h As Long
    Dim col As Long
    Dim datarow As Long
    Dim source As String

    ' Get the persisted record set
    source    =    Environ("LocalAppData")    &    "\KESoftware\
Reports\ecatalogue\xmldata.xml"
    Set RecordSet = New ADODB.RecordSet
    RecordSet.Open source, "Provider=MSPersist"

    ' Get the active page to send the data to
    Set Worksheet = ThisWorkbook.ActiveSheet
    Application.Visible = True
```

```
    ' Put out all of the column headers
    col = 1
    ListColumnNames Worksheet, RecordSet, col

    ' Print out all the row data
    While Not RecordSet.EOF
        col = 1
        datarow = datarow + 1
        For h = 0 To RecordSet.Fields.count - 1
                Worksheet.Cells(datarow + 1, col).Value =
RecordSet.Fields(h).Value
                col = col + 1
        Next
        RecordSet.MoveNext
    Wend

    Worksheet.Range("A1").CurrentRegion.Select
    Worksheet.Columns.AutoFit
    Set RecordSet = Nothing
End Sub

Private Sub ListColumnNames(ByVal ws As Excel.Worksheet, ByVal rs
As ADODB.RecordSet, ByRef col As Long)
    Dim i As Long
    ' Loop through the record set pulling out the column names
    For i = 0 To rs.Fields.count - 1
            ws.Cells(1, col).Value = rs.Fields(i).Name
            col = col + 1
    Next
End Sub
```



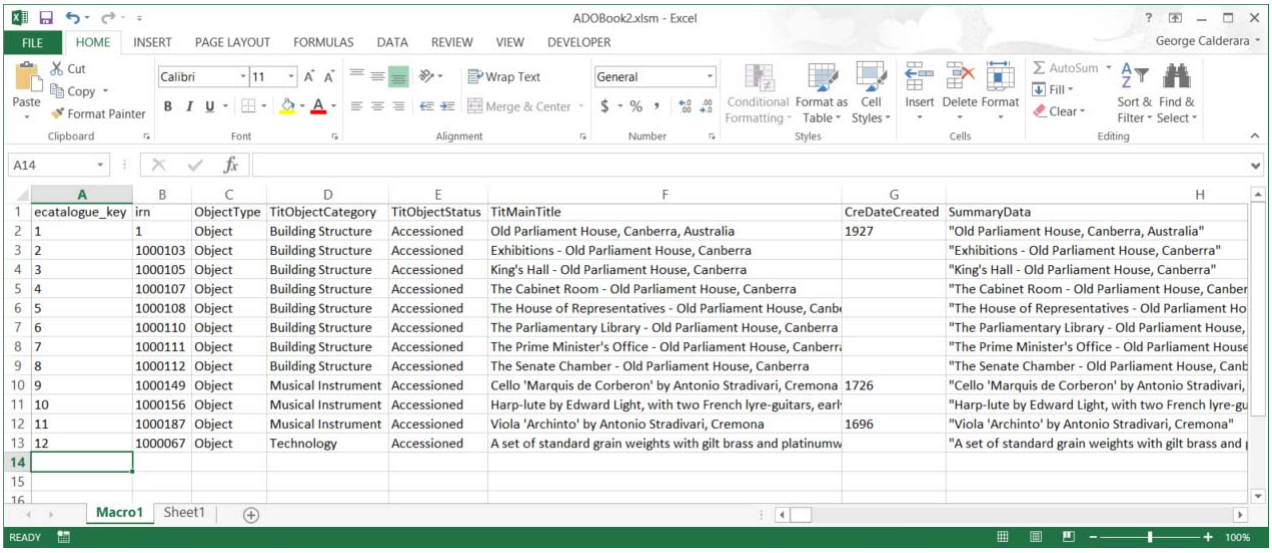
3. Double-click **ThisWorbook** in the VBAProject pane and copy and paste the following code:

```
Sub Workbook_Open()
' Load up the ADO File
Sheet1.OpenAdoFile
End Sub
```



4. Save the report and upload it to your EMu report (page 24) on the Report Type tab of the Report Properties box.

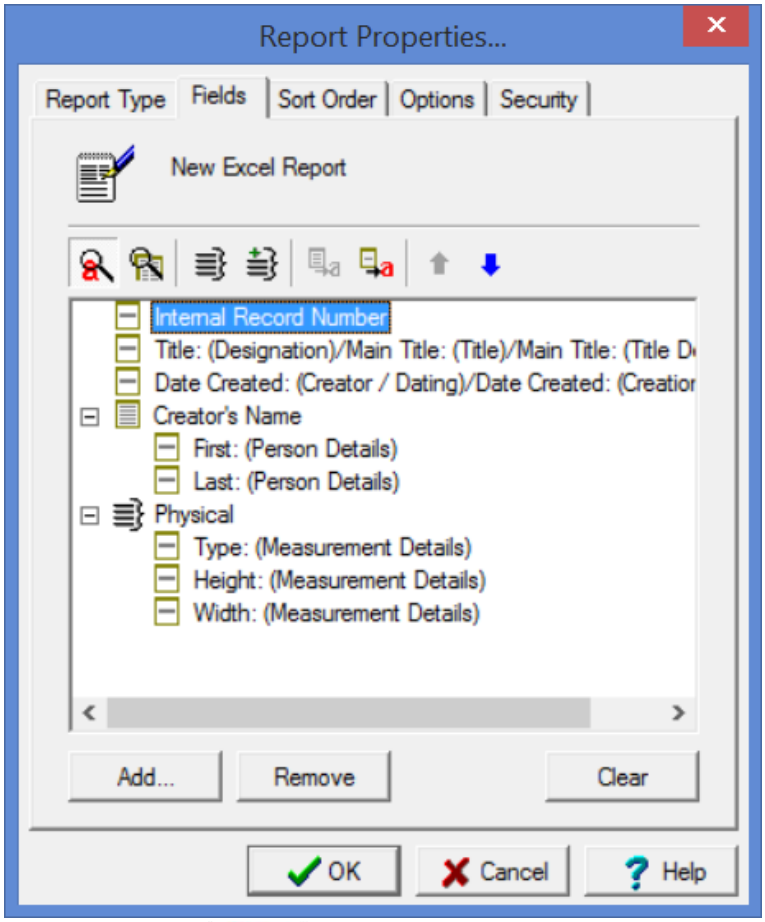
When the report is run in EMu, an Excel report is generated:



	A	B	C	D	E	F	G	H
1	ecatalogue_key	irn	ObjectType	TitObjectCategory	TitObjectStatus	TitMainTitle	CreDateCreated	SummaryData
2	1	1	Object	Building Structure	Accessioned	Old Parliament House, Canberra, Australia	1927	"Old Parliament House, Canberra, Australia"
3	2	1000103	Object	Building Structure	Accessioned	Exhibitions - Old Parliament House, Canberra		"Exhibitions - Old Parliament House, Canberra"
4	3	1000105	Object	Building Structure	Accessioned	King's Hall - Old Parliament House, Canberra		"King's Hall - Old Parliament House, Canberra"
5	4	1000107	Object	Building Structure	Accessioned	The Cabinet Room - Old Parliament House, Canberra		"The Cabinet Room - Old Parliament House, Canber
6	5	1000108	Object	Building Structure	Accessioned	The House of Representatives - Old Parliament House, Canb		"The House of Representatives - Old Parliament Ho
7	6	1000110	Object	Building Structure	Accessioned	The Parliamentary Library - Old Parliament House, Canberra		"The Parliamentary Library - Old Parliament House,
8	7	1000111	Object	Building Structure	Accessioned	The Prime Minister's Office - Old Parliament House, Canberra		"The Prime Minister's Office - Old Parliament House
9	8	1000112	Object	Building Structure	Accessioned	The Senate Chamber - Old Parliament House, Canberra		"The Senate Chamber - Old Parliament House, Canb
10	9	1000149	Object	Musical Instrument	Accessioned	Cello 'Marquis de Corberon' by Antonio Stradivari, Cremona	1726	"Cello 'Marquis de Corberon' by Antonio Stradivari,
11	10	1000156	Object	Musical Instrument	Accessioned	Harp-lute by Edward Light, with two French lyre-guitars, earl		"Harp-lute by Edward Light, with two French lyre-gu
12	11	1000187	Object	Musical Instrument	Accessioned	Viola 'Archinto' by Antonio Stradivari, Cremona	1696	"Viola 'Archinto' by Antonio Stradivari, Cremona"
13	12	1000067	Object	Technology	Accessioned	A set of standard grain weights with gilt brass and platinumw		"A set of standard grain weights with gilt brass and
14								
15								
16								

How to create an Excel Report with nested tables using the ADO RecordSet

1. Repeat Step1: Create a new report in EMu (page 24).
   For this example, the following fields were selected. Note the two nested tables - *Creator's Name* and *Physical*:



2. In Excel, click 

on the Developers tab.
3. Double-click **Sheet1** in the VBAProject pane:
4. Copy and paste the following VB code:

```
Sub Read_XML_Data()

    Dim rst As ADODB.Recordset
    Dim Worksheet As Excel.Worksheet
    Dim i As Long
    Dim j As Long
    Dim source As String
    Dim datarow As Long
    Dim saverow As Long
    Dim lastrow As Long
    Dim col As Long

    ' These next declaration is a little odd. Its needed in cases where the entire value
    ' of a nested table is blank. In these cases it is necessary to force a number of columns to be skipped when printing
    ' out field values. Oddly, as long as a nested table has at least one value, then there is no issue.
    ' There is only a need to declare one variable for each nested table.
    ' In this example there are only two nested tables so two declarations are needed
    ' The value assigned to each variable will depend on the number of fields in that nested table.
    ' In this example the first nested table is the CreCreatorRef_tab, which has two fields, i.e. NamFirst and NamLast
    ' and the second nested table, i.e Physical, has 3 fields, i.e. PhyType, PhyHeight and PhyWidth

    Dim firstnestedtable As Long
    Dim secondnestedtable As Long
    Dim nestedtablecount As Long

    firstnestedtable = 2
    secondnestedtable = 3
    nestedtablecount = 1

    ' Get the persisted record set
    source = Environ("LocalAppData") & "\KESoftware\Reports\ecatalogue\xmldata.xml"
    Set rst = New ADODB.Recordset
    rst.Open source, "Provider=MSPersist"
```

```
' Get the active page to send the data to
Set Worksheet = ThisWorkbook.ActiveSheet
Application.Visible = True

'Add column labels
Worksheet.Cells(1, 1).Select
ActiveCell.EntireRow.Insert
Worksheet.Cells(1, 1).Value = "Record No"
Worksheet.Cells(1, 2).Value = "IRN No"
Worksheet.Cells(1, 3).Value = "Title"
Worksheet.Cells(1, 4).Value = "Date Created"
Worksheet.Cells(1, 5).Value = "Creator First"
Worksheet.Cells(1, 6).Value = "Creator Last"
Worksheet.Cells(1, 7).Value = "Physical Type"
Worksheet.Cells(1, 8).Value = "Physical Length"
Worksheet.Cells(1, 9).Value = "Physical Width"

col = 1
' Start printing data from Row 3
datarow = 3
lastrow = datarow
While Not rst.EOF
    col = 1

    If datarow < lastrow Then
        datarow = lastrow
    End If

    For j = 0 To rst.Fields.Count - 1
        If rst.Fields(j).Type = adChapter Then
            If rst.Fields(j).Value.BOF Then
                Worksheet.Cells(datarow, col).Value = ""
                If nestedtablecount = 1 Then
                    col = col + firstnestedtable
                    nestedtablecount = nestedtablecount + 1
                ElseIf nestedtablecount = 2 Then
                    col = col + secondnestedtable
                    nestedtablecount = nestedtablecount + 1
                End If
            Else
                If rst.Fields(j).Value.EOF Then
                    Worksheet.Cells(datarow, col).Value = ""
                    If nestedtablecount = 1 Then
```

```
                            col = col + firstnestedtable
                            nestedtablecount = nestedtablecount
+ 1
                        ElseIf nestedtablecount = 2 Then
                            col = col + secondnestedtable
                            nestedtablecount = nestedtablecount
+ 1
                        End If
                    Else
                        saverow = datarow
                        ListNestedValues              Worksheet,
rst.Fields(j).Value,    col,    datarow,    lastrow,    saverow,
nestedtablecount
                    End If
                End If
            Else
                If IsNull(rst.Fields(j).Value) Then
                    Worksheet.Cells(datarow, col).Value = ""
                Else
                    Worksheet.Cells(datarow,    col).Value    =
rst.Fields(j).Value
                End If
                col = col + 1
            End If
        Next
        rst.MoveNext
        datarow = datarow + 1
        nestedtablecount = 1
    Wend

    'Closing the recordset.
    rst.Close

    'Release object from memory.

    Worksheet.Range("A1").CurrentRegion.Select
    Worksheet.Columns.AutoFit
    Set rst = Nothing

End Sub

Private Sub ListNestedValues(ByVal ws As Excel.Worksheet, ByVal
rs As ADODB.Recordset, ByRef col As Long, ByRef datarow As Long,
ByRef   lastrow   As   Long,   ByRef   saverow   As   Long,   ByRef
```

```
nestedtablecount As Long)
    Dim i As Long
    Dim j As Long
    Dim startrow As Long

    ' Loop through a nested table pulling out the row values
    j = 0
    startrow = saverow
    While Not rs.EOF
        max = 1
        j = col
        For i = 0 To rs.Fields.Count - 1
            ' Don't print key values
            If   rs.Fields(i).Name   <>   "ecatalogue_key"   And
rs.Fields(i).Name <> "CreCreatorRef_key" And rs.Fields(i).Name <>
"Physical_key" _
            Then
                If IsNull(rs.Fields(i).Value) Then
                    ws.Cells(startrow + 1, j).Value = ""
                    j = j + 1
                Else
                    If rs.Fields(i).Type = adChapter Then
                        ListNestedValues ws, rs.Fields(i).Value,
j, datarow, lastrow, saverow, nestedtablecount
                        datarow = startrow
                    Else
                        ws.Cells(startrow,      j).Value      =
rs.Fields(i).Value
                        j = j + 1
                    End If
                End If
            End If
        Next
        rs.MoveNext
        startrow = startrow + 1
    Wend

    If (j > 0) Then
        col = j
    End If

    If startrow > lastrow Then
        lastrow = startrow
    End If
```

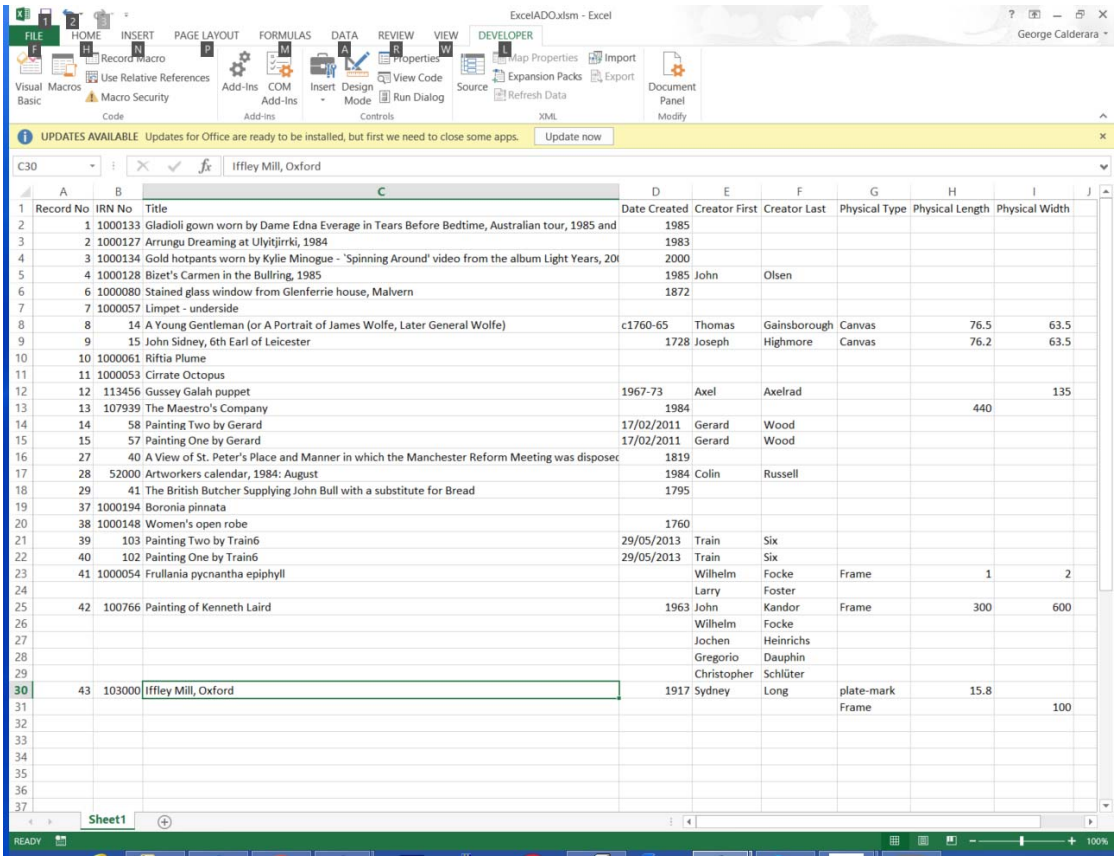
```
        nestedtablecount = nestedtablecount + 1
    End Sub
```

5. Double-click **ThisWorbook** in the VBAProject pane and copy and paste the following code:

```
Sub Workbook_Open()
' Load up the ADO File
Sheet1.Read_XML_Data
End Sub
```

6. Save the report and upload it to your EMu report (page 24) on the Report Type tab of the Report Properties box.

When the report is run in EMu, an Excel report is generated:



Record No	IRN No	Title	Date Created	Creator First	Creator Last	Physical Type	Physical Length	Physical Width
1	1000133	Gladioli gown worn by Dame Edna Everage in Tears Before Bedtime, Australian tour, 1985 and	1985					
2	1000127	Arrungu Dreaming at Ulyitjirrki, 1984	1983					
3	1000134	Gold hotpants worn by Kylie Minogue - 'Spinning Around' video from the album Light Years, 20(	2000					
4	1000128	Bizet's Carmen in the Bullring, 1985	1985	John	Olsen			
6	1000080	Stained glass window from Glenferrie house, Malvern	1872					
7	1000057	Limpet - underside						
8	14	A Young Gentleman (or A Portrait of James Wolfe, Later General Wolfe)	c1760-65	Thomas	Gainsborough	Canvas	76.5	63.5
9	15	John Sidney, 6th Earl of Leicester	1728	Joseph	Highmore	Canvas	76.2	63.5
10	1000061	Riftia Plume						
11	1000053	Cirrate Octopus						
12	113456	Gussey Galah puppet	1967-73	Axel	Axelrad			135
13	107939	The Maestro's Company	1984				440	
14	58	Painting Two by Gerard	17/02/2011	Gerard	Wood			
15	57	Painting One by Gerard	17/02/2011	Gerard	Wood			
27	40	A View of St. Peter's Place and Manner in which the Manchester Reform Meeting was disposec	1819					
28	52000	Artworkers calendar, 1984: August	1984	Colin	Russell			
29	41	The British Butcher Supplying John Bull with a substitute for Bread	1795					
37	1000194	Boronia pinnata						
38	1000148	Women's open robe	1760					
39	103	Painting Two by Train6	29/05/2013	Train	Six			
40	102	Painting One by Train6	29/05/2013	Train	Six			
41	1000054	Frullania pycnantha epiphyll		Wilhelm	Focke	Frame	1	2
				Larry	Foster			
42	100766	Painting of Kenneth Laird	1963	John	Kandor	Frame	300	600
				Wilhelm	Focke			
				Jochen	Heinrichs			
				Gregorio	Dauphin			
				Christopher	Schlüter			
43	103000	Iffley Mill, Oxford	1917	Sydney	Long	plate-mark	15.8	
						Frame		100

SECTION 4

Registry entries

The Type Registry entry indicates which export type to use for each report request.

The format of this Registry entry is ;

`System|Setting|Reports|Type|Crystal CSV|`*value*

value is `0` or `1`:

`0` Generates data in the existing format.

`1` Generates data in the new Crystal ODBC format.

If this entry is not present, a *value* of `0` is assumed.

`System|Setting|Reports|Type|Crystal ADO|`*value*

value is `0` or `2`:

`0` Generates data in the existing format.

`2` Generates data in the new Crystal ADO record set.

If this entry is not present, a *value* of `0` is assumed.

`System|Setting|Reports|Type|Microsoft ADO|`*value*

where:

value is `0` or `3`:

`0` Generates data in the existing format.

`3` Generates data in the new Microsoft ADO format.

If this entry is not present, a *value* of `0` is assumed.

Index