

WALK THROUGH

STEP 1 – CREATE THE STORED PROCEDURE

My report will have 2 parameters (ServiceTimeLine & FormCode). The trick on creating the report is for your stored procedure (or sprocs) to handle all scenarios given the parameters involved.

So for example, if you have 2 parameters, you would have to handle the ff. in the sproc:

- Parameter 1 – Null, Parameter 2 – Entered
- Parameter 1 – Entered, Parameter 2 – Entered
- Parameter 1 – Entered, Parameter 2 – Null
- Parameter 1 – Null, Parameter 2 – Null

In my report, I decided only to handle my report scenarios where the user only enters the ff. information:

- ServiceTimeLine – Entered, FormCode – Entered
- ServiceTimeLine – Entered, FormCode – Null

That means, all other scenarios left will give me no data to display at all (blank report).

Here's my SPROC code. I have the IF..ELSE statement to handle the report scenarios I mentioned above.

```
CREATE PROCEDURE sp_RPTAcctInstructions
    @ServiceTimeLine varchar(7),
    @FormCode varchar(4)
AS
IF @FormCode is null
    BEGIN
        SELECT
            O.FormCode,
            A.AcctName,
            M.AcctRetailer,
            O.PreCall,
            O.VRUTimeLine,
            O.OrderID,
            O.PreCallInst,
            O.Instructions,
            O.Attachments,
            O.ProductDesc,
            O.ProjectGoals,
            O.VRUQuestions
        FROM (Orders as O
        INNER JOIN Account as A ON O.FormCode=A.FormCode)
```

```

INNER JOIN MainStore as M ON A.MainStoreID=M.MainStoreID
WHERE
    O.VRUTimeLine = @ServiceTimeLine
ORDER BY
    O.FormCode
END
ELSE
BEGIN
SELECT
    O.FormCode,
    A.AcctName,
    M.AcctRetailer,
    O.PreCall,
    O.VRUTimeLine,
    O.OrderID,
    O.PreCallInst,
    O.Instructions,
    O.Attachments,
    O.ProductDesc,
    O.ProjectGoals,
    O.VRUQuestions
FROM (Orders as O
INNER JOIN Account as A ON O.FormCode=A.FormCode)
INNER JOIN MainStore as M ON A.MainStoreID=M.MainStoreID
WHERE
    O.VRUTimeLine = @ServiceTimeLine
    and O.Formcode = @FormCode
ORDER BY
    O.FormCode
END

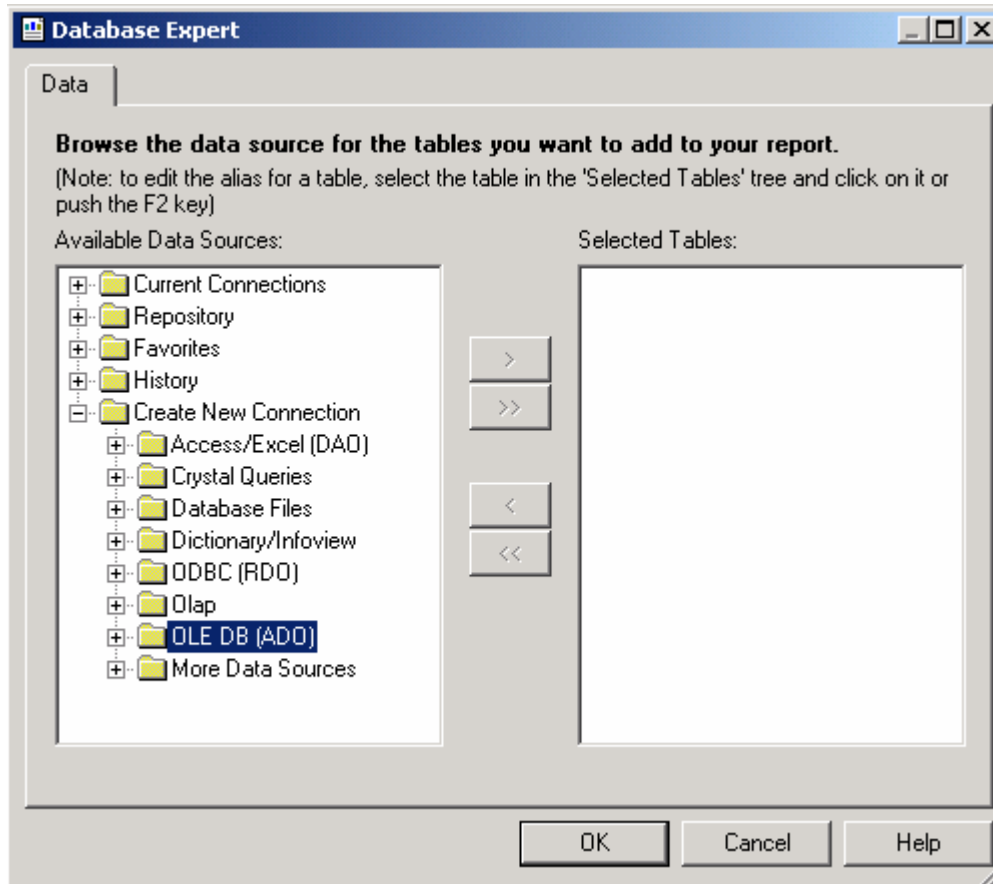
```

STEP 2 – DESIGN YOUR REPORT IN CRYSTAL

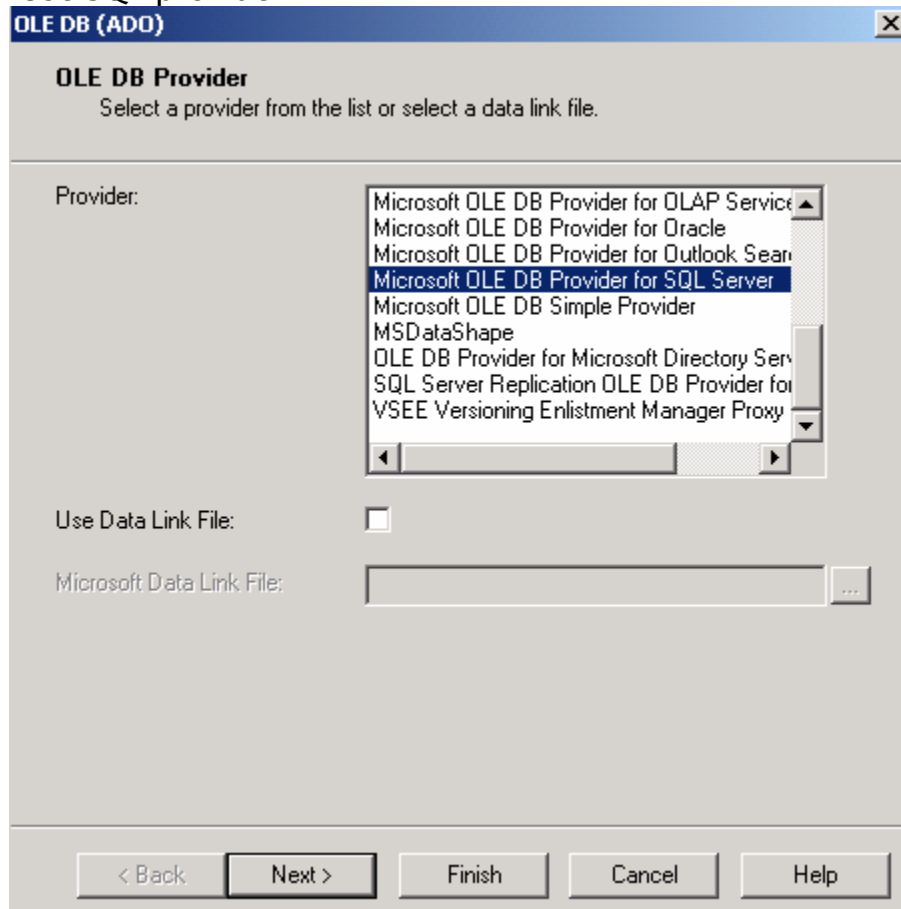
I design and call all my CR report outside VB6 and/or VB.NET. One advantage of this is easier maintenance of the reports. If I need to modify something on the report, I do it directly so I don't need to recompile the VB executable. BTW, this walk though is for CR9 Developer Edition.

- (1) Open CR
- (2) Create a new blank report

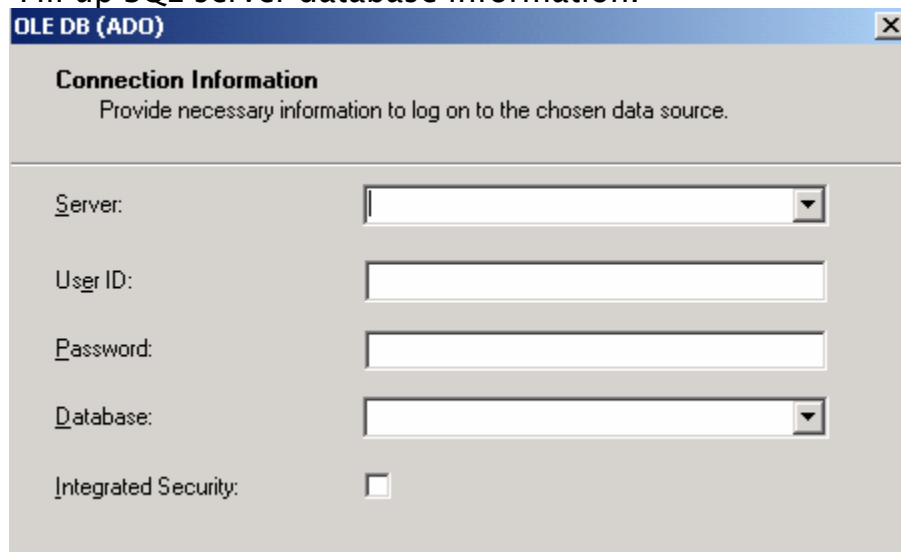
(3) On the Database Expert use OLE DB



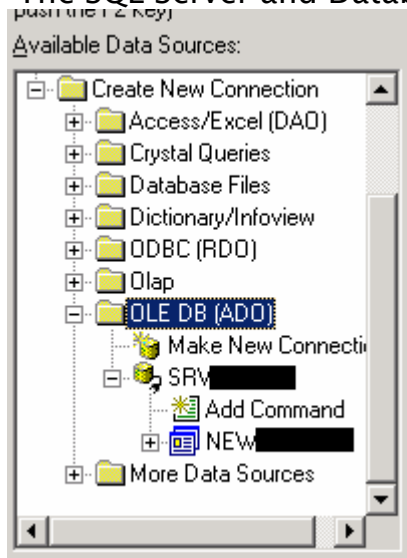
(4) Use SQL provider.



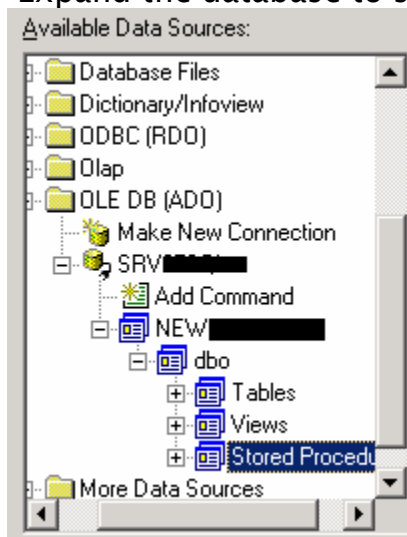
(5) Fill up SQL server database information.




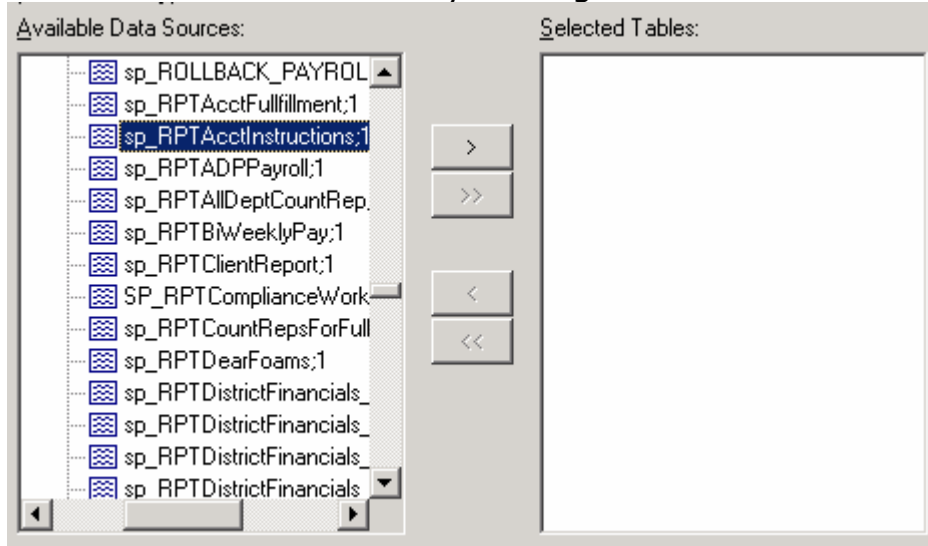
(6) The SQL Server and Database will be visible as a data source after.



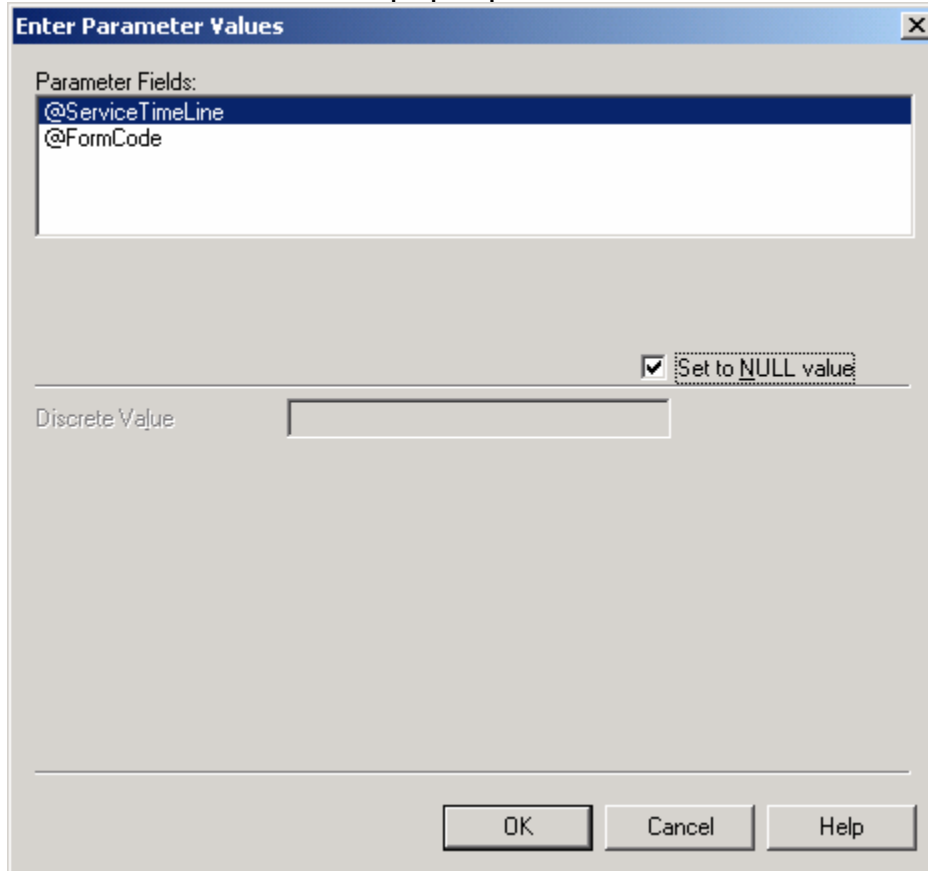
(7) Expand the database to see Tables, Views and Store Procedure



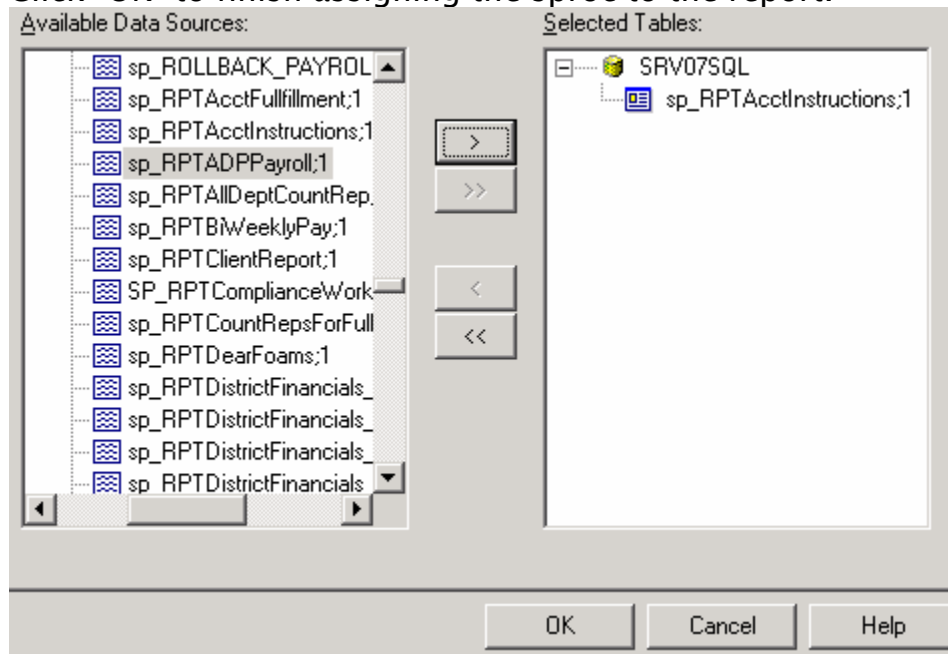
- (8) Expand the Stored Procedure, locate and highlight the SPROC the add to the Selected Tables by clicking on the  button and



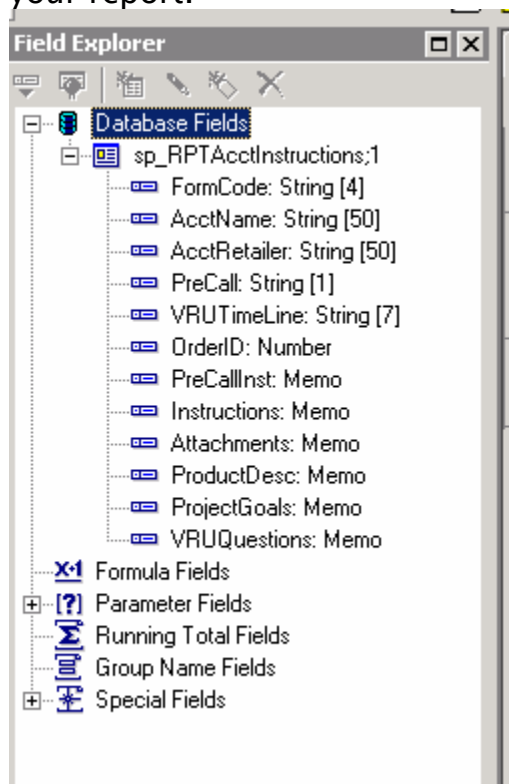
- (9) You would then see this pop-up window. Click on "OK".



- (10) You would see the sproc added on the Selected Table pane. Click “OK” to finish assigning the sproc to the report.



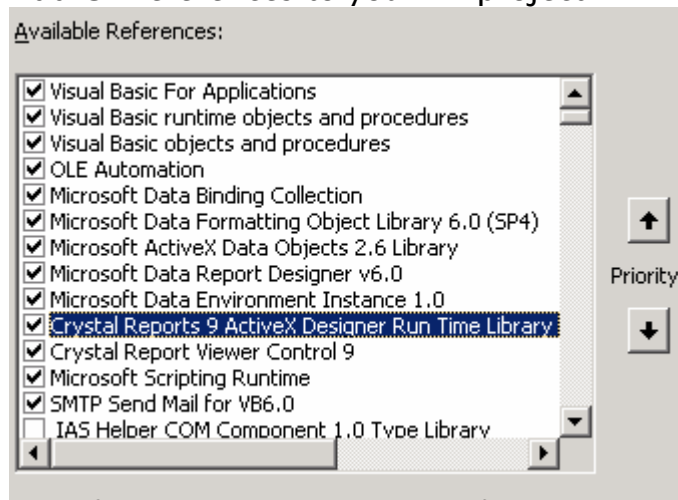
- (11) Your Field Explorer will have the sproc field as reference for your report.



(12) Start designing and testing your report.

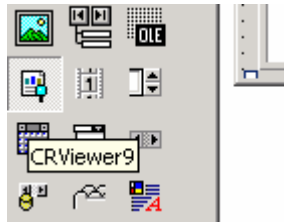
STEP 3 – CALL YOUR REPORT IN VB

(1) Add CR references to your VB project.



(2) Add a form into your VB project.

(3) Add the CRVIEWER control in the form.



(4) Here's a sample VB code needed in the report form. You may have a BUTTON on another form to launch or call this report form (I used a grid).

Option Explicit

```
Dim crxApplication As New CRAXDRT.Application  
Dim crxReport As CRAXDRT.Report
```

```
Private Sub Form_Load()  
    Set crxApplication = New CRAXDRT.Application  
    Set crxReport = crxApplication.OpenReport(REPORT_DIR) ← full path + name of report  
    '-THIS IS FOR THE MAIN REPORT  
    With crxReport.Database.Tables(1).ConnectionProperties  
        .Item("Provider") = "SQLOLEDB"  
        .Item("Data source") = "SRVXXXXX"  
        .Item("Initial Catalog") = "NEWXXXXXX"  
        .Item("User ID") = "XXXXXX"  
        .Item("Password") = "XXXXXX"
```

```

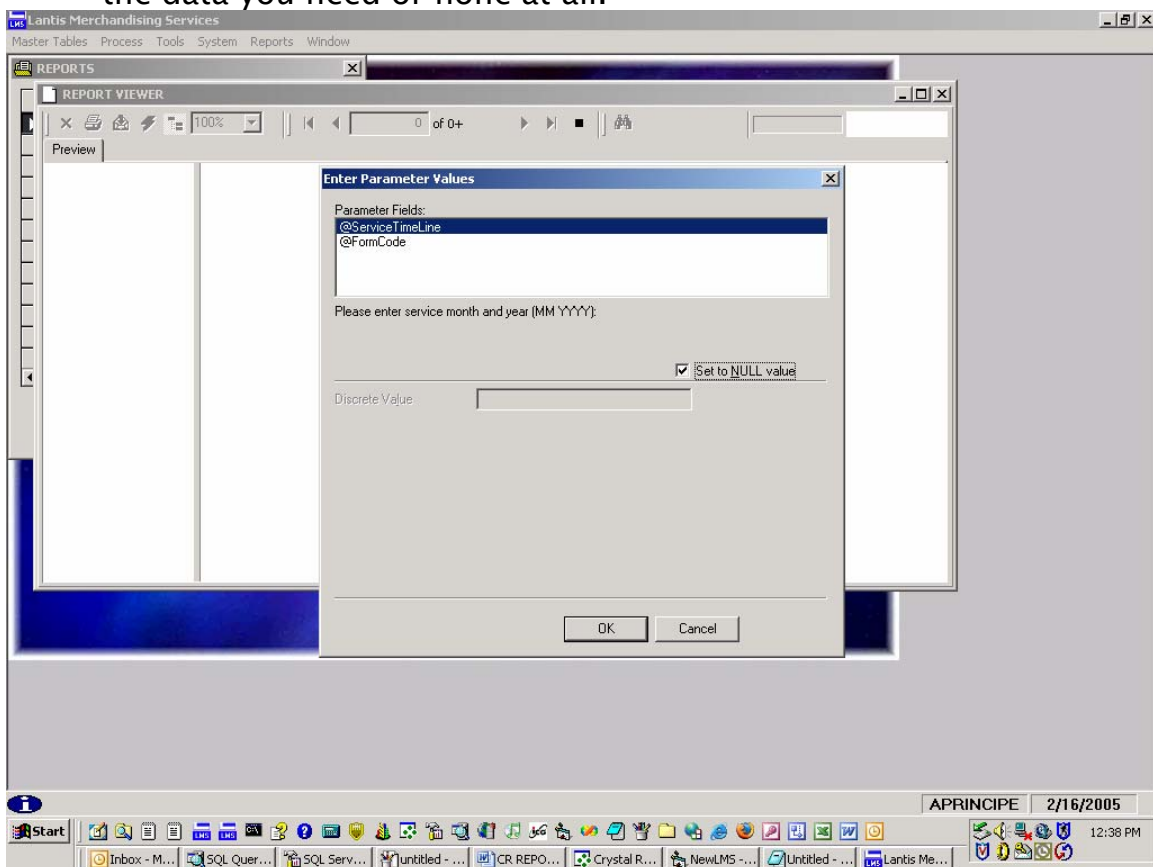
End With
CRViewer91.ReportSource = crxReport
CRViewer91.ViewReport
End Sub

Private Sub Form_Resize()
    With CRViewer91
        .Top = 0
        .Left = 0
        .Width = Me.ScaleWidth
        .Height = Me.ScaleHeight
    End With
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set crxReport = Nothing
    Set crxApplication = Nothing
End Sub

```

- (5) Once the report form is launched, the parameters on the sproc will be prompted. Again, depending upon how you coded your sproc to handle the different parameter entries, the report may bring back the data you need or none at all.



AGAIN, THIS WALK THROUGH USES VB6 + CR9 DEVELOPER EDITION + SQL SERVER. THIS MAY OR MAY NOT !!! WORK FOR YOU BUT I HOPE IT WOULD LEAD YOU TO THE RIGHT DIRECTION.