

Re: XML binding advice

Source: <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.xml/2004-03/0200.html>

From: Brooks Stoner [MSFT] (*brookss_at_online.microsoft.com*)

Date: 03/09/04

Date: Tue, 9 Mar 2004 10:37:20 -0800

Eric,

It sounds like you are looking for a review of your code and some feedback on how it could be improved. If so, then there is one thing that you should consider before anything else. Instead of building your SQL statement in code and executing it, you should build a stored procedure and just pass the parameters into it from your code prior to executing it. This has a number of advantages, including being a better performing solution, as well as being more secure.

Performance – By writing the SQL statement as a stored procedure, SQL Server can create an execution plan when it compiles the procedure, and that execution plan will take into account any indexes you have on the tables being queried. If you pass a statement from code, SQL Server will have to compile it and compute the execution plan each time it runs, both of which take time. Using this method also eliminates the need to build a large Select statement in your code, which will improve your application performance and memory usage. By concatenating strings together the way you are, you are forcing the CLR to instantiate string objects in memory for each piece you add to the string. For future reference, if you need to build a large string by adding a lot of smaller strings, you should use the `StringBuilder` class (it is much more performant and uses less memory than concatenating string objects).

Security – If you create the SQL statement in code, the account you use to connect to the database must have select permissions on the tables you are querying. If you use a stored procedure, you can provide execute permissions on the stored procedure without granting any permissions to the underlying tables. Another argument for this is that by building a statement as a string in your application, you could be subject to a SQL injection attack (the string values you read contain code that could change the query you are building).

Stored procedure –

Create Procedure BuldReportResults

 @StartDate DATETIME,

 @EndDate DATETIME,

 @OfficeCode VARCHAR(50)

microsoft.public.dotnet.xml: Re: XML binding advice

```
as
begin
    SELECT ...
end
```

I hope this is what you were looking for.

Brooks

```
--
This posting is provided "AS IS" with no warranties, and confers no rights.
Use of included script samples are subject to the terms specified at
http://www.microsoft.com/info/copyright.htm.
"Eric" <eric.tuomikoski@kutakrock.com> wrote in message
news:00B2039A-951C-4C93-8164-6DA5D83F131D@microsoft.com...
> I generate an SQL statement, fill a dataset with the results, then run an
write XML doc from the dataset, then bind the XML to the Crystal report.
When someone changes a parameter and runs the procedure again, it recreates
everything again. Sounds slow. Please see my procedure below and comment. I
let the user build their reports requirements from a form, then send the
values to this Function. This is my first Dot Net try, so Im sure it isnt
pretty. Thanks All, Eric. Code below.
>
> Public Function ProcessExpense(ByVal StartDate As Date, ByVal EndDate As
Date, ByVal XMLPath As String, ByVal GroupOnOffice As Boolean, ByVal
Officeselectd As String)
>     Dim sqlExpense As String
>     Dim adoOleDbConnection As OleDbConnection
>     Dim adoOleDbDataAdapter As OleDbDataAdapter
>     Dim dataSet As DataSet
>     Dim connectionString As String = ""
>     Dim crDTEReportOffice As New DTEExpenseOffice
>     Dim crDTEReportName As New DTEExpenseName
>     StartDate = Format(StartDate, "MM/dd/yyyy")
>     EndDate = Format(EndDate, "MM/dd/yyyy")
>
>     connectionString = "Provider=SQLOLEDB;"
>     connectionString += "Server=dt0101;Database=dte2002;"
>     connectionString += "User ID=sa;Password="
>
>     'Create and open a connection using the connection string
>     adoOleDbConnection = New OleDbConnection(connectionString)
>
>     'Expense(SQL)
>     sqlExpense = "SELECT '" & StartDate & "' as StartDate,'" & EndDate
& "' as EndDate,timekeepers.name AS Name, dtetime.timekeeper AS TimeKeeper#,
(CASE timekeepers.office "
>     sqlExpense = sqlExpense & "WHEN '01' THEN 'Omaha' "
>     sqlExpense = sqlExpense & "WHEN '02' THEN 'Denver' "
>     sqlExpense = sqlExpense & "WHEN '03' THEN 'Washington DC' "
>     sqlExpense = sqlExpense & "WHEN '04' THEN 'Atlanta' "
>     sqlExpense = sqlExpense & "WHEN '05' THEN 'Scottsdale' "
>     sqlExpense = sqlExpense & "WHEN '06' THEN 'Pasadena' "
>     sqlExpense = sqlExpense & "WHEN '09' THEN 'Des Moines' "
>     sqlExpense = sqlExpense & "WHEN '10' THEN 'Arkansas' "
>     sqlExpense = sqlExpense & "WHEN '11' THEN 'Kansas City' "
>     sqlExpense = sqlExpense & "WHEN '12' THEN 'Wichita' "
>     sqlExpense = sqlExpense & "WHEN '14' THEN 'Richmond' "
>     sqlExpense = sqlExpense & "WHEN '18' THEN 'Chicago' "
>     sqlExpense = sqlExpense & "WHEN '17' THEN 'Irvine' "
```

Re: XML binding advice

microsoft.public.dotnet.xml: Re: XML binding advice

```
>         sqlExpense = sqlExpense & "WHEN '19' THEN 'Lincoln' "
>         sqlExpense = sqlExpense & "WHEN '21' THEN 'Oklahoma City' "
>         sqlExpense = sqlExpense & "WHEN '97' THEN 'National' "
>         sqlExpense = sqlExpense & "ELSE timekeepers.office end)AS Office,
"
>         sqlExpense = sqlExpense & "dtetime.isexpense, CONVERT(decimal(10,
2), dtetime.hours) AS Amount, dtetime.client AS Client, dtetime.matter AS
Matter, "
>         sqlExpense = sqlExpense &
"convert(varchar(50),dtetime.workdate,101) AS WorkDate "
>         sqlExpense = sqlExpense & "FROM           dtetime INNER JOIN "
>         sqlExpense = sqlExpense & "timekeepers ON dtetime.timekeeper =
timekeepers.timekeeper "
>         sqlExpense = sqlExpense & "WHERE           (dtetime.isexpense = 'Y') and
timekeepers.timekeeper <> '22222' and timekeepers.timekeeper <> '22223' and
timekeepers.timekeeper <> '22224'and timekeepers.timekeeper <> '22225'and
timekeepers.timekeeper <> '22226'and timekeepers.timekeeper <> '22227'and
timekeepers.timekeeper <> '22231' "
>         sqlExpense = sqlExpense & "and timekeepers.timekeeper <> '22230' "
>         sqlExpense = sqlExpense & "and timekeepers.timekeeper <> '22228' "
>         sqlExpense = sqlExpense & "and timekeepers.timekeeper <> '22229' "
>         If Officeselected = "timekeepers.office" Then
>             sqlExpense = sqlExpense & "and timekeepers.office =
timekeepers.office "
>             Else
>                 sqlExpense = sqlExpense & "and timekeepers.office = '" &
Officeselected & "'"
>             End If
>         sqlExpense = sqlExpense & " and dtetime.released = 'N' "
>         sqlExpense = sqlExpense & "and workdate between '" & StartDate &
"' and '" & EndDate & "' ORDER BY DATEPART(yyyy, workdate), DATEPART(mm,
workdate), DATEPART(dd, workdate)"
>
>         'Retrieve the data using the SQL statement and existing connection
>         adoOleDbDataAdapter = New OleDbDataAdapter(sqlExpense,
adoOleDbConnection)
>         'Create a instance of a Dataset
>         dataSet = New DataSet
>
>         'Fill the dataset with the data retrieved. The name of the table
>         'in the dataset must be the same as the table name in the report.
>
>         Try
>             adoOleDbDataAdapter.Fill(dataSet, "DTEExpense")
>
>         Catch objectDataFill As Exception
>             'Throw objectDBConnect
>
>             MsgBox(objectDataFill.Message)
>
>         End Try
>
>         'Create an instance of the strongly-typed report object
>
>         dataSet.WriteXml(XMLPath, XmlWriteMode.WriteSchema)
>         'Pass the populated dataset to the report
>         If GroupOnOffice = True Then
>             crDTEReportOffice.SetDataSource(dataSet)
>                 ProcessExpense = crDTEReportOffice
>
>         Else
>             crDTEReportName.SetDataSource(dataSet)
```

microsoft.public.dotnet.xml: Re: XML binding advice

```
>         ProcessExpense = crDTEReportName
>     End If
>
>     ''Set the viewer to the report object to be previewed.
>     adoOleDbConnection.Close()
>     adoOleDbConnection = Nothing
```