

نحوه Login کردن به SQL Server در VS.NET 2005

زمانی که شما از پایگاه داده SQL Server استفاده می کنید باید از NameSpace CrystalDecisions.Shared استفاده گردد. برای Connect شدن به SQL Server کافی است که خصلت های ServerName, DatabaseName, UserID, Password را تنظیم نمایید چنانچه خاصیت ServerName را تنظیم نکنید از ServerName گزارشان استفاده خواهد نمود. برای تنظیم مسیر گزارشان به دو صورت زیر عمل می کنیم.

- **WebForm**: برای پروژه های وب از دستور `Server.MapPath()` استفاده می

```
string reportPath = Server.MapPath("Report1.rpt ");
```

کنیم.

- **WinForm**: برای پروژه های ویندوز از دستور `Application.StartupPath`

استفاده می کنیم.

```
string RepPath=Application.StartupPath + @"\"Report\Report1.rpt";
```

دستورات بالا فقط برای مشخص نمودن مسیر گزارشهای طراحی شده توسط شما مورد استفاده قرار می گیرد حال برای ربط دادن این گزارشات به Viewer کریستال رپورت می توانید از دستور زیر استفاده نمایید.

- **WebForm** و **WinForm**: برای پروژه های وب و ویندوز کافی است که خاصیت

`crystalReportViewer.ReportSource` را تنظیم نمایید.

```
crystalReportViewer.ReportSource = RepPath;
```

حال در این قسمت به تنظیم `ConnectionString` خواهیم پرداخت. در واقع

`ConnectionString` اطلاعات مربوط به `Connection` از قبیل

`UserID, DataBaseName, Password` را تنظیم می نماید.

WebForm و **WinForm**:

```
ConnectionString connectionInfo = new ConnectionInfo();
connectionInfo.ServerName = "DynamicMethods";
connectionInfo.DatabaseName = "Db1";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";
```

حال برای ربط دادن `ConnectionString` به `crystalReportViewer` از تابع زیر استفاده می کنیم.

WebForm و WinForm:

```
private void SetLogin(ConnectionString connectionInfo)
{
    TableLogOnInfos TbLogOnInfos = crystalReportViewer.LogOnInfo;
    foreach (TableLogOnInfo TbLogOnInfos in TbLogOnInfos)
    {
        TbLogOnInfos.ConnectionString = connectionInfo;
    }
}
```

پس برای یک پروژه ویندوز (WinForm) در C# داریم:

```
ConnectionString connectionInfo = new ConnectionInfo();
connectionInfo.ServerName = "DynamicMethods";
connectionInfo.DatabaseName = "Db1";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";
string RepPath=Application.StartupPath + @"\Report\Report1.rpt";
crystalReportViewer.ReportSource = RepPath;
SetLogin(connectionInfo);
```

```
-----
private void SetLogin(ConnectionString connectionInfo)
{
    TableLogOnInfos TbLogOnInfos = crystalReportViewer.LogOnInfo;
    foreach (TableLogOnInfo TbLogOnInfos in TbLogOnInfos)
    {
        TbLogOnInfos.ConnectionString = connectionInfo;
    }
}
```

دستورات بالا را در `Page_Load` قرار دهید.

برای یک پروژه وب (WebForm) در C# داریم:

```
ConnectionString connectionInfo = new ConnectionInfo();
connectionInfo.ServerName = "DynamicMethods";
connectionInfo.DatabaseName = "Db1";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";
string reportPath = Server.MapPath("Report1.rpt");
crystalReportViewer.ReportSource = RepPath;
SetLogin(connectionInfo);
```

```
-----
private void SetLogin(ConnectionString connectionInfo)
{
    TableLogOnInfos TbLogOnInfos = crystalReportViewer.LogOnInfo;
    foreach (TableLogOnInfo TbLogOnInfos in TbLogOnInfos)
    {
        TbLogOnInfos.ConnectionString = connectionInfo;
    }
}
```

دستورات بالا را در `Page_Init` قرار دهید.

نحوه Login کردن به SQL Server بوسیله SQL Authentication

در این قسمت شما با استفاده از مسائل امنیتی SQL Server گزارشی را نمایش خواهید داد. برای این منظور از شیء ReportDocument استفاده خواهیم کرد. زمانی که به یک جدول متصل می شویم یک سطح امنیتی بوسیله SQL Server برای آن جدول ساخته می شود برای کنترل این سطح امنیتی می توانید از خاصیت ConnectionInfo استفاده نمایید.

برای تعریف شیء ReportDocument به صورت زیر عمل می کنیم:

```
ReportDocument RepDoc=new ReportDocument ();
```

برای تنظیم مسیر گزارشتان به دو صورت زیر عمل می کنیم.

- **WebForm**: برای پروژه های وب از دستور `Server.MapPath()` استفاده می

```
string reportPath = Server.MapPath("Report1.rpt");
```

- **WinForm**: برای پروژه های ویندوز از دستور `Application.StartupPath` استفاده می کنیم.

```
string RepPath=Application.StartupPath + @"Report\Report1.rpt";
```

دستورات بالا فقط برای مشخص نمودن مسیر گزارشهای طراحی شده توسط شما مورد استفاده قرار می گیرد حال برای ربط دادن این گزارشات به Viewer کریستال رپورت می توانید از دستور زیر استفاده نمایید. برای این کار از شیء ReportDocument ای را که ایجاد کرده ایم استفاده می نمایم.

```
RepDoc.Load(RepPath);
```

حال شیء ReportDocument را به Viewer کریستال رپورت ربط می دهیم.

```
crystalReportViewer.ReportSource = RepDoc;
```

بعد از اعمال تنظیمات ConnectionInfo که در قسمت قبلی توضیح داده شد به نوشتن کد تابع

SetLogin می پردازیم.

```
private void SetLogin(ConnectionInfo connectionInfo, ReportDocument reportdocument)
{
    Tables tables = reportdocument.Database.Tables;
    foreach (CrystalDecisions.CrystalReports.Engine.Table table in tables)
    {
        TableLogOnInfo TbLogonInfo = table.LogOnInfo;
        TbLogonInfo.ConnectionInfo = connectionInfo;
        table.ApplyLogOnInfo (TbLogonInfo);
    }
}
```

پس برای یک پروژه ویندوز (WinForm) در C# داریم:

```
ConnectionInfo connectionInfo = new ConnectionInfo();
ReportDocument RepDoc=new ReportDocument();
connectionInfo.ServerName = "DynamicMethods";
connectionInfo.DatabaseName = "Db1";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";
string RepPath=Application.StartupPath + @"\Report\Report1.rpt";
RepDoc.Load(RepPath);
crystalReportViewer.ReportSource = RepDoc;
SetLogin(connectionInfo, RepDoc);
-----
private void SetLogin(ConnectionInfo connectionInfo,ReportDocument
reportdocument)
{
    Tables tables = reportdocument.Database.Tables;
    foreach (CrystalDecisions.CrystalReports.Engine.Table table in
tables)
    {
        TableLogOnInfo TbLogonInfo = table.LogOnInfo;
        TbLogonInfo.ConnectionInfo = connectionInfo;
        table.ApplyLogOnInfo (TbLogonInfo);
    }
}
```

برای یک پروژه وب (WebForm) در C# داریم:

```
ConnectionInfo connectionInfo = new ConnectionInfo();
ReportDocument RepDoc=new ReportDocument();
connectionInfo.ServerName = "DynamicMethods";
connectionInfo.DatabaseName = "Db1";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";
string reportPath = Server.MapPath("Report1.rpt");
RepDoc.Load(RepPath);
crystalReportViewer.ReportSource = RepDoc;
SetLogin(connectionInfo, RepDoc);
-----
private void SetLogin(ConnectionInfo connectionInfo,ReportDocument
reportdocument)
{
    Tables tables = reportdocument.Database.Tables;
    foreach (CrystalDecisions.CrystalReports.Engine.Table table in
tables)
    {
        TableLogOnInfo TbLogonInfo = table.LogOnInfo;
        TbLogonInfo.ConnectionInfo = connectionInfo;
        table.ApplyLogOnInfo (TbLogonInfo);
    }
}
```

نحوه کار با پارامترها

برای کار کردن با پارامترهای کریستال رپورت بوسیله VS.NET بایستی به ثورت زیر عمل نمایید (البته به صورت کلی در مورد پارامترها در آموزش های قبلی توضیح دادیم).
برای ارسال مقادیر به پارامترهای کریستال رپورت بوسیله VS.NET کافی است که از کد زیر استفاده نمایید.

```
ReportDocument RepDoc = new ReportDocument();
string RepPath = Application.StartupPath + "\\CrystalReport1.rpt";
RepDoc.Load(RepPath);
crystalReportViewer1.ReportSource = RepDoc;
//-----

ParameterValues parameterValues = new ParameterValues();
ParameterDiscreteValue parameterDiscreteValue = new
ParameterDiscreteValue();
parameterDiscreteValue.Value = "ParameterValue";
parameterValues.Add(parameterDiscreteValue);
ParameterFieldDefinitions parameterFieldDefinitions =
RepDoc.DataDefinition.ParameterFields;
ParameterFieldDefinition parameterFieldDefinition =
parameterFieldDefinitions["ParameterName"];
parameterFieldDefinition.ApplyCurrentValues(parameterValues);
```

محدود کردن داده ها در گزارش بوسیله Selection Formula

بیشتر گزارشات دارای یک محدوده خاص می باشد، برای مثال شما می خواهید گزارش فاکتورهای خرید مربوط به مشتری X (کد مشتری X عدد 1050 می باشد) را داشته باشید برای این منظور گزارش شما باید محدود شود و این کار توسط Selection Formula انجام می شود. برای ساخت گزارش بالا از دستورات زیر استفاده نمایید.

```
ReportDocument RepDoc = new ReportDocument();
string RepPath = Application.StartupPath + "\\CrystalReport1.rpt";
RepDoc.Load(RepPath);

string SFormula= "{Customer.Code}=1050";
RepDoc.DataDefinition.RecordSelectionFormula = SFormula;
crystalReportViewer1.ReportSource = RepDoc;
```

حال اگر کد مشتری از نوع String باشد:

```
ReportDocument RepDoc = new ReportDocument();
string RepPath = Application.StartupPath + "\\CrystalReport1.rpt";
RepDoc.Load(RepPath);

string SFormula= "{Customer.Code}='1050'";
RepDoc.DataDefinition.RecordSelectionFormula = SFormula;
crystalReportViewer1.ReportSource = RepDoc;
```

برای کنترل کدهای نوشته شده می توانید از Select Expert در خود کریستال رپورت استفاده نموده و نگاهی به کدهای تولید شده برای استفاده بیشتر بکنید.

نحوه Connect شدن به SubReport ها

در بیشتر گزارشاتی که توسط Crystal Report طراحی می شود در بعضی مواقع پیش می آید که از SubReport ها استفاده نمایم. ولی چنانچه اینگونه گزارشات را در یک کامپیوتر دیگر اجرا نمایم باید SubReport را به DataBase متصل نمایم.

برای اینکار کافی است که مراحل زیر را انجام دهید.

1. ConnectionInfo را تنظیم کنید.

```
ConnectionInfo connectionInfo = new ConnectionInfo();
connectionInfo.ServerName = ".";
connectionInfo.DatabaseName = "anbar";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";
```

2. مسیر گزارش را مشخص می کنیم

```
ReportDocument RepDoc = new ReportDocument();
string RepPath = Application.StartupPath + "\\CrystalReport1.rpt";
RepDoc.Load(RepPath);
```

3. برای Report اصلی یک تابع Login می نویسیم

```
private void SetLoginMainReport(ConnectionInfo connectionInfo,
ReportDocument reportdocument)
{
    Tables tables = reportdocument.Database.Tables;
    foreach (CrystalDecisions.CrystalReports.Engine.Table table
in tables)
    {
        TableLogOnInfo TbLogonInfo = table.LogOnInfo;
        TbLogonInfo.ConnectionInfo = connectionInfo;
        table.ApplyLogOnInfo(TbLogonInfo);
    }
}
```

4. حال همین کار را برای SubReport انجام می دهیم.

```
private void SetLoginSubReport(ConnectionInfo connectionInfo,
ReportDocument reportdocument)
{
    Sections sections = reportdocument.ReportDefinition.Sections;
    foreach (Section section in sections)
    {
        ReportObjects RepObjs = section.ReportObjects;
        foreach (ReportObject RepObj in RepObjs)
        {
            if (RepObj.Kind == ReportObjectKind.SubreportObject)
            {
                SubreportObject SubreportObj = (SubreportObject)RepObj;
                ReportDocument SubReportDoc =
                SubreportObj.OpenSubreport(SubreportObj.SubreportName);
                SetLoginMainReport(connectionInfo, SubReportDoc);
            }
        }
    }
}
```

5. حال دو تابع نوشته شده را فراخوانی می کنیم این فراخوانی در کل به صورت زیر است.

```
ConnectionInfo connectionInfo = new ConnectionInfo();
connectionInfo.ServerName = ".";
connectionInfo.DatabaseName = "anbar";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";
ReportDocument RepDoc = new ReportDocument();
string RepPath = Application.StartupPath + "\\CrystalReport1.rpt";
RepDoc.Load(RepPath);
SetLoginMainReport(connectionInfo, RepDoc);
SetLoginSubReport(connectionInfo, RepDoc);
crystalReportViewer1.ReportSource=RepDoc;
```


نحوه اتصال گزارش به DataSet

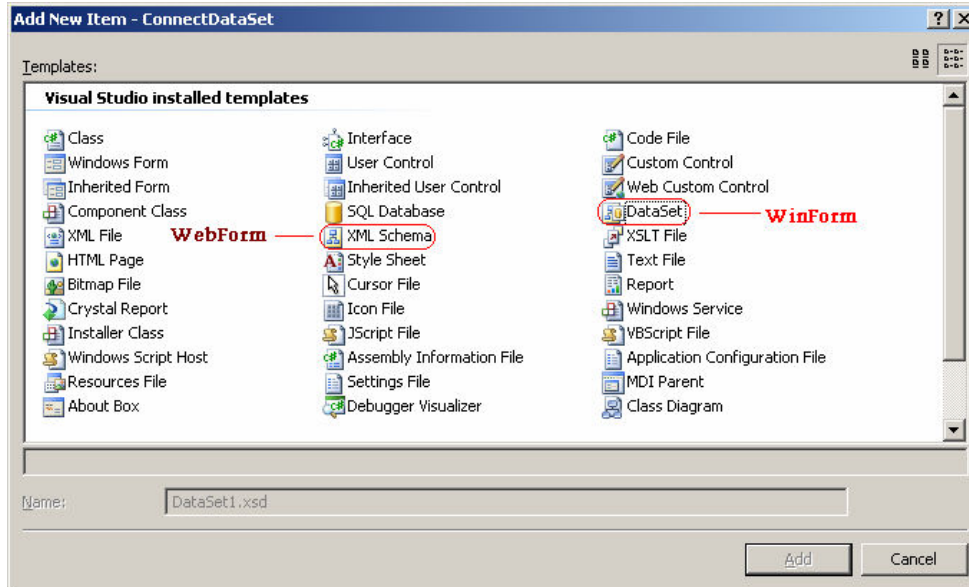
در این قسمت با چگونگی اتصال گزارشتان به DataSet آشنا خواهید شد البته این کار توسط DataSet Schema انجام خواهد شد. اینگونه گزارشات مستقیماً به DataBase متصل نمی شود و این کار توسط Schema انجام می گیرد. توجه داشته باشید که اینگونه گزارشات برای پروژه هایی می باشد که از شی DataSet استفاده می نمایند.

برای شروع یک Connection از نوع DataSet Schema خواهیم ساخت برای اینکار
1. بر روی Server Explorer راست کلیک نموده و گزینه Add Connection را

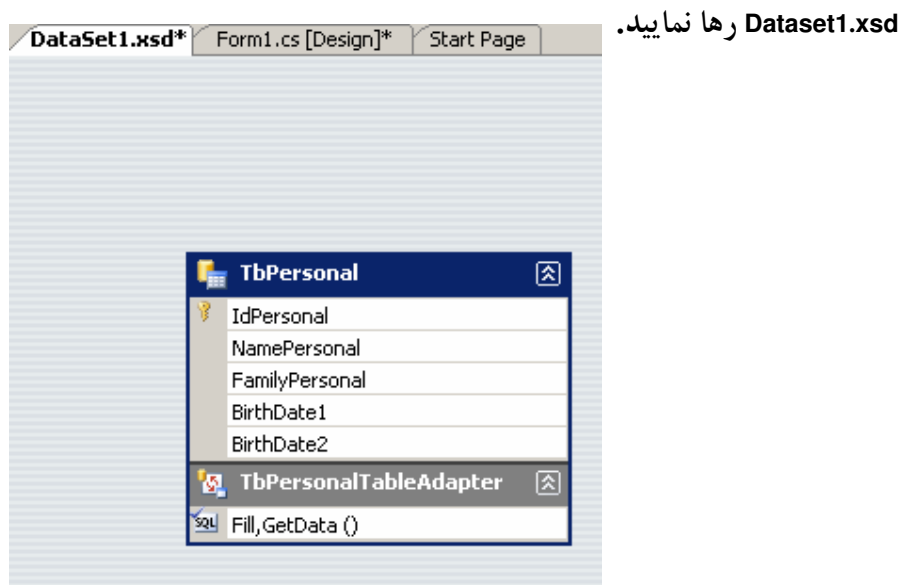
انتخاب نمایید.

2. سپس تنظیمات مربوطه را انجام دهید.

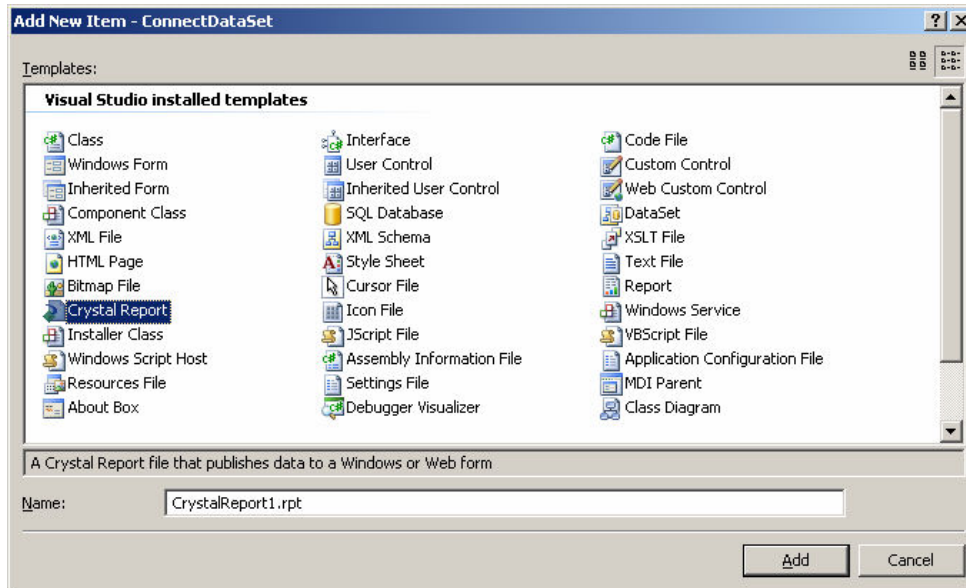
3. حال بر روی پروژه در Solution Explorer کلیک راست نموده و از منوی Add گزینه New Item را انتخاب و از فرم مربوطه برای پروژه های ویندوز DataSet را انتخاب و برای پروژه های وب XML Schema را انتخاب نمایید.



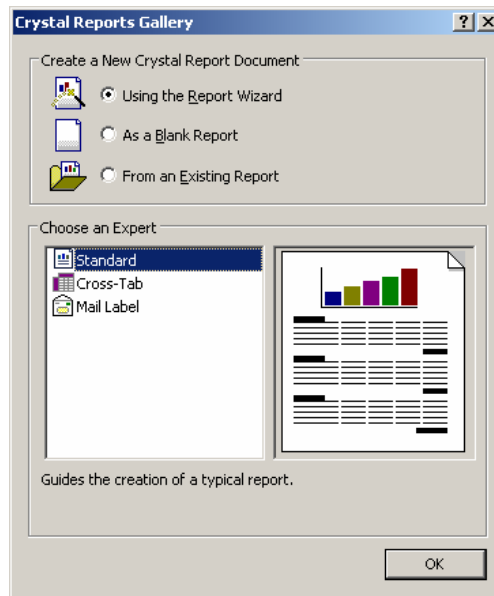
4. حال از قسمت Solution Explore جدول مربوطه را انتخاب نمایید و بر روی



5. حال بر روی پروژه در Solution Explorer کلیک راست نموده و از منوی Add گزینه New Item را انتخاب و از فرم مربوطه Crystal Report را انتخاب نمایید.



6. همانند شکل زیر با استفاده از Wizard گزارشی را بوسیله Schema خواهیم ساخت.

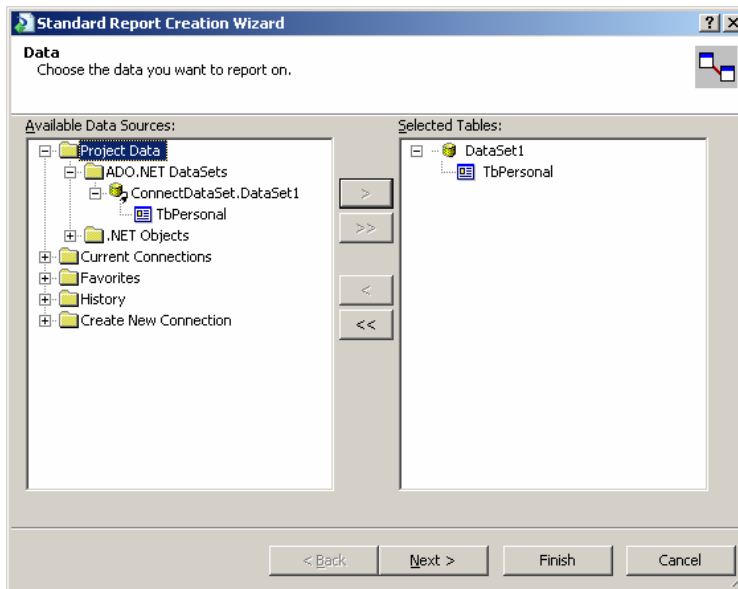


7. در فرم Standard Report Creation Wizard از قسمت Available Data

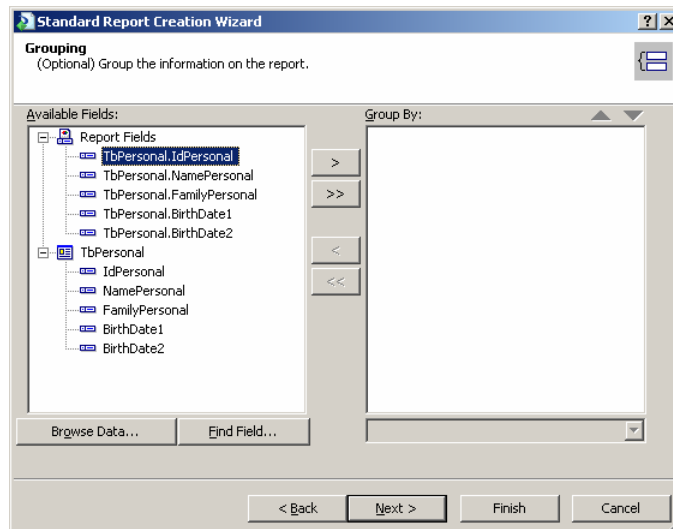
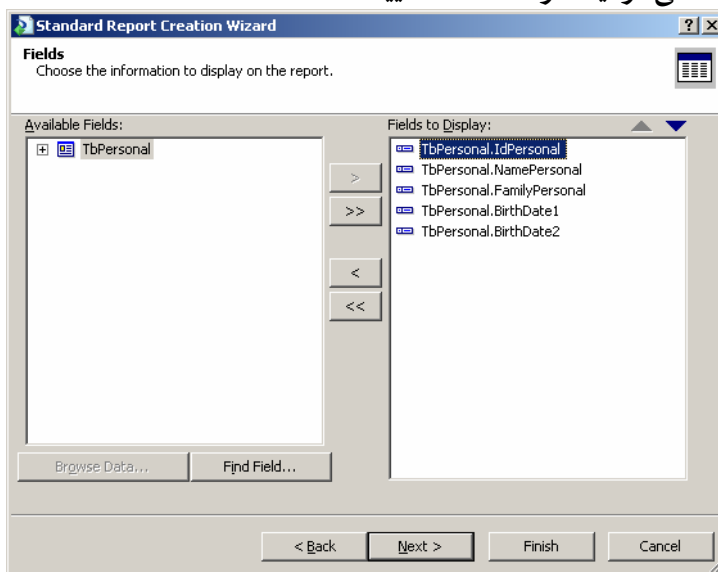
Source که به صورت tree می باشد، از

Project Data-> ADO.NET DataSet -> YourDataSetName-> Your Table Name

در صفحه بعد شما می توانید تصویر فرم مربوطه را مشاهده نمایید.



8. بروی **Next** کلیک نمایید تا فرم های بعدی را نیز مشاهده نمایید. مراحل را تا رسیدن به **Finish** می توانید خود مشاهده نمایید.



9. سپس کد مربوطه را در پروژه استفاده نمایید.

```
string ConStr = "Data Source=.;Initial Catalog=Anbar;User
ID=sa;Password=123456";
SqlConnection SQLCon = new SqlConnection(ConStr);
SqlDataAdapter SQLAdapter = new SqlDataAdapter("SELECT * FROM
TbPersonal", SQLCon);
DataSet1 dataset = new DataSet1();
SQLAdapter.Fill(dataset, "TbPersonal");
//-----

ConnectionInfo connectionInfo = new ConnectionInfo();
ReportDocument RepDoc = new ReportDocument();
connectionInfo.ServerName = ".";
connectionInfo.DatabaseName = "anbar";
connectionInfo.UserID = "sa";
connectionInfo.Password = "123456";

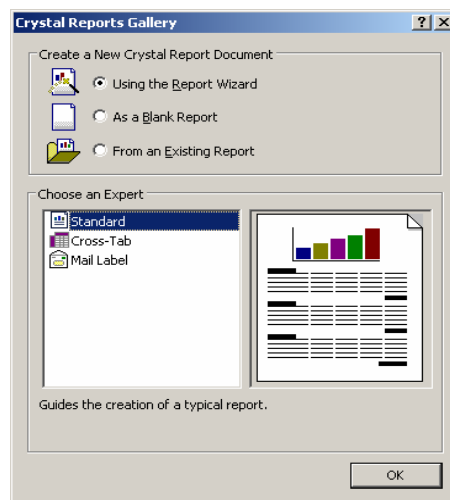
string RepPath = Application.StartupPath + @"\CrystalReport1.rpt";
RepDoc.Load(RepPath);
RepDoc.SetDataSource(dataset);
crystalReportViewer1.ReportSource = RepDoc;
```

نحوه اتصال گزارش به IDataReader

بدون شک در بیشتر پروژه های NET. از شیء DataReader به مراتب استفاده می گردد زیرا سرعت آن به مراتب بیشتر از DataSet می باشد. همانطور که می دانید شیء DataReader خود نیز از IDataReader (IDataReader خود یک Interface می باشد) استفاده می نماید. در این قسمت می خواهیم به کمی از برنامه نویسی Multi Tier استفاده کنیم برای این منظور شما در Solution Explorer خود باید دو پروژه ایجاد نمایید که پروژه اول از نوع Windows Application و پروژه دوم از نوع Class Library باید باشد. سپس در پروژه دوم (Class Library) کد زیر را برای Calss1.cs بنویسید.

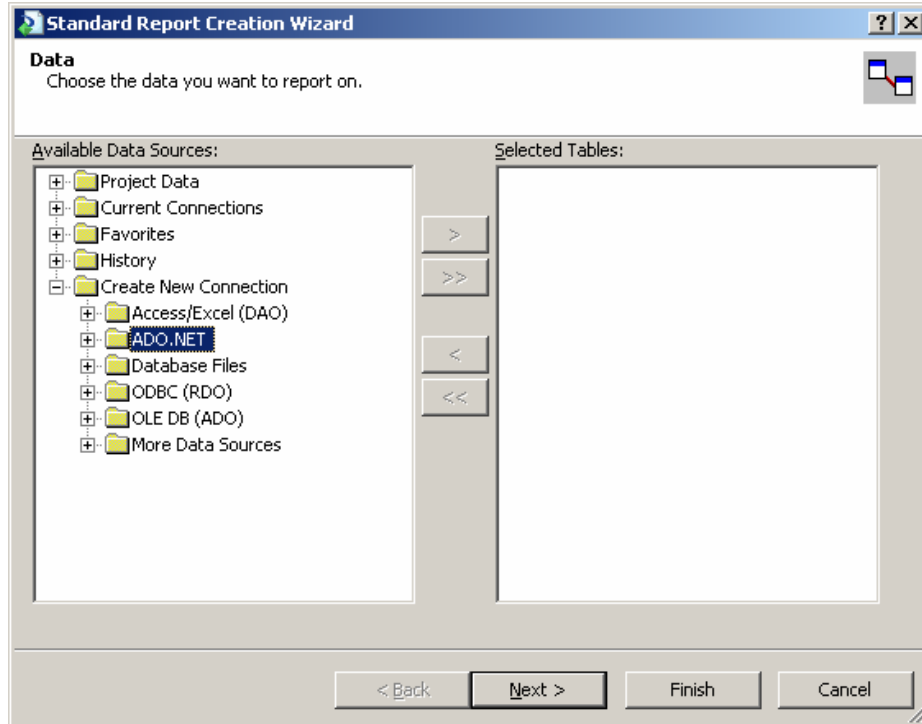
```
private const string ConStr = "Data Source=.;Initial
Catalog=Anbar;User ID=sa;Password=123456";
public static IDataReader GetTbPersonal()
{
    SqlConnection SQLCon = new SqlConnection(ConStr);
    SQLCon.Open();
    SqlCommand SQLCommand = new SqlCommand("SELECT * FROM
TbPersonal", SQLCon);
    IDataReader iDataR = SQLCommand.ExecuteReader();
    return iDataR;
}
```

1. بر روی پروژه دوم (Class Library) کلیک راست نموده و گزینه Build را انتخاب نمایید. مشاهده می نمایید که برای Build شما یک DLL ساخته شده است. مسیر DLL را از پنجره Output به خاطر بسپارید (یعنی کپی کنید).
2. بر روی پروژه اول (Application) کلیک راست نموده و یک آیتم Crystal Report را انتخاب نمایید.
3. همانند شکل زیر با استفاده از Wizard گزارش را خواهیم ساخت.

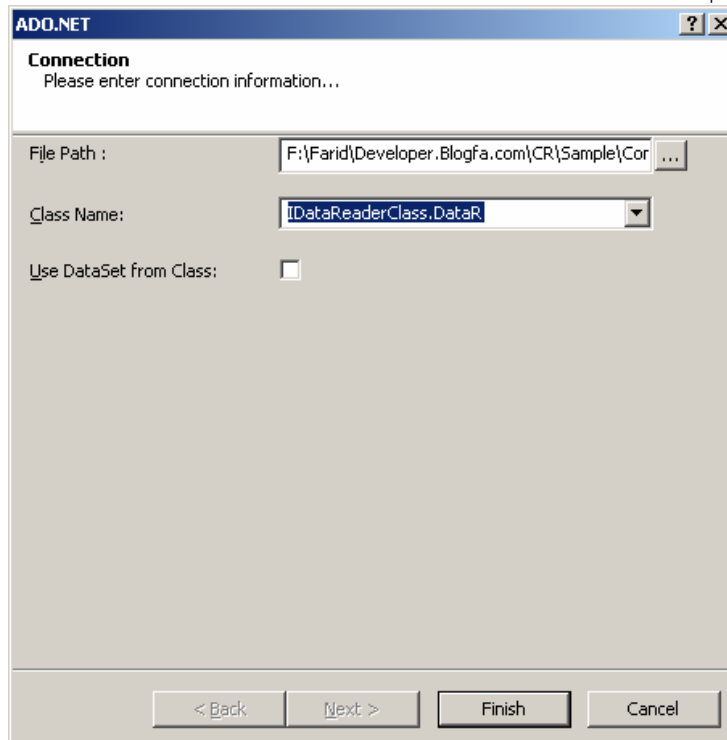


4. در فرم Standard Report Creation Wizard از قسمت Available Data Source که به صورت tree می باشد، Node مربوط به ساختن Connection جدید را انتخاب نمایید.

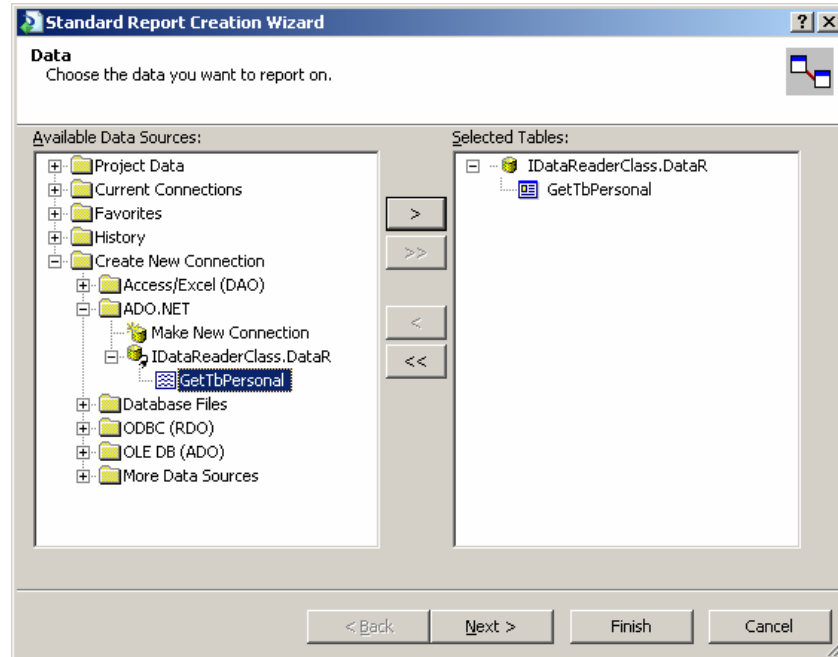
Create New Connection -> ADO.NET



5. حال از فرم ADO.NET فایل DLL را انتخاب نمایید.



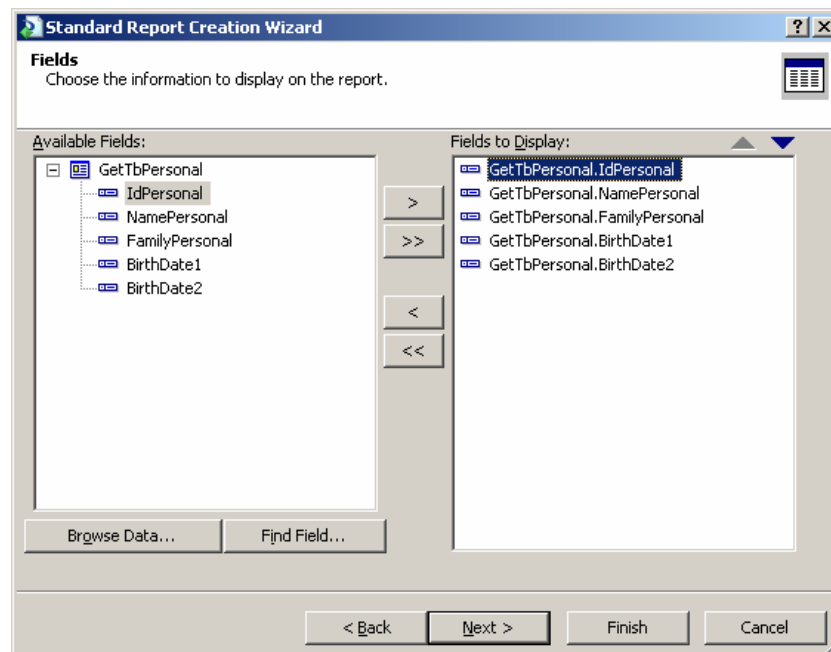
6. بر روی کلید **Finish** کلیک نموده و همانند تصویر زیر عمل نمایید.



7. در شکل زیر مشاهده می نمایید که به چه صورت فیلدهایی که ما خواسته ایم را برای ما از **Class** استخراج نموده است.

`SqlCommand SQLCommand = new SqlCommand("SELECT * FROM TbPersonal", SQLCon);`
 در واقع خط بالا از کد نمایانگر تعداد فیلدهای استخراجی است (* یعنی **ALL**) حال چنانچه کد به صورت زیر باشد

`SqlCommand SQLCommand = new SqlCommand("SELECT ID,Name FROM TbPersonal", SQLCon);`
 فقط دو فیلد **ID,Name** را برای ما نمایش می دهد.



8. حال کد زیر را برای پروژه اول (Application) بنویسید تا گزارستان نمایش داده

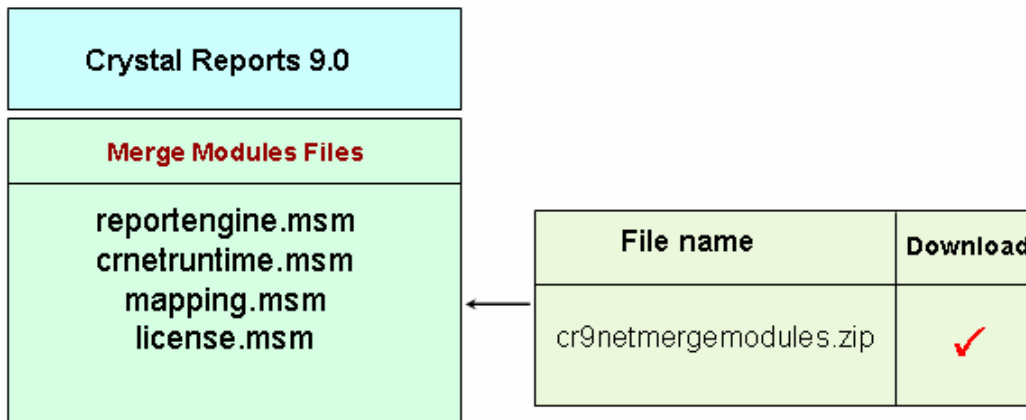
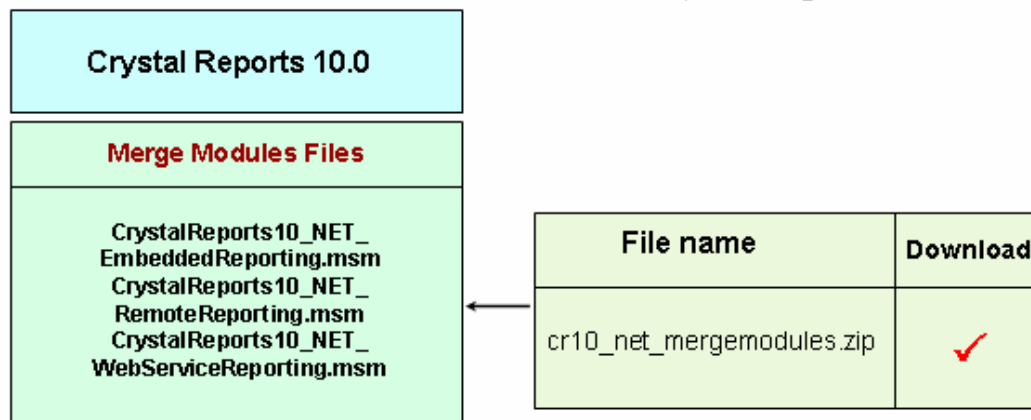
شود.

```
string RepPath = Application.StartupPath + "\\CrystalReport1.rpt";  
ReportDocument RepDoc = new ReportDocument();  
RepDoc.Load(RepPath);  
crystalReportViewer1.ReportSource = RepDoc;
```

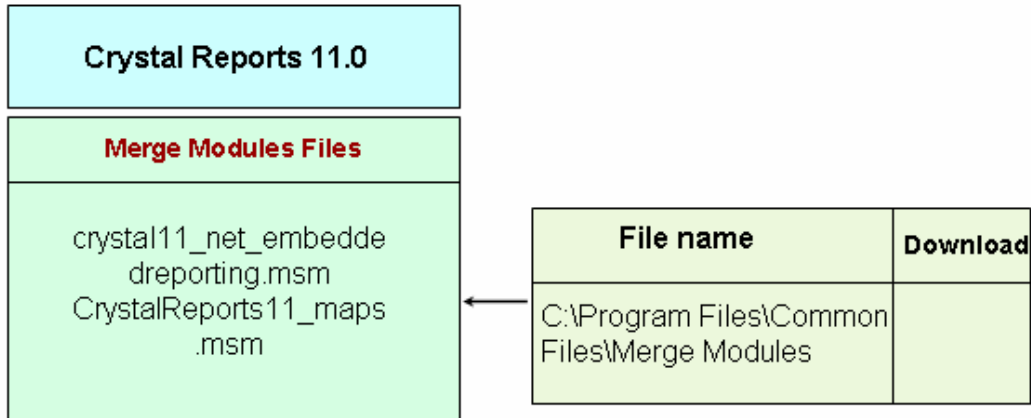
این روش یکی از روش های جالب برای برنامه نویسی کریستال رپورت می باشد.
بهبتر است روی این روش زیاد کار کنید تا درک لازم را برای کارهای بیشتر با این روش داشته
باشید.

مروری بر Merge Module های Crystal Reports

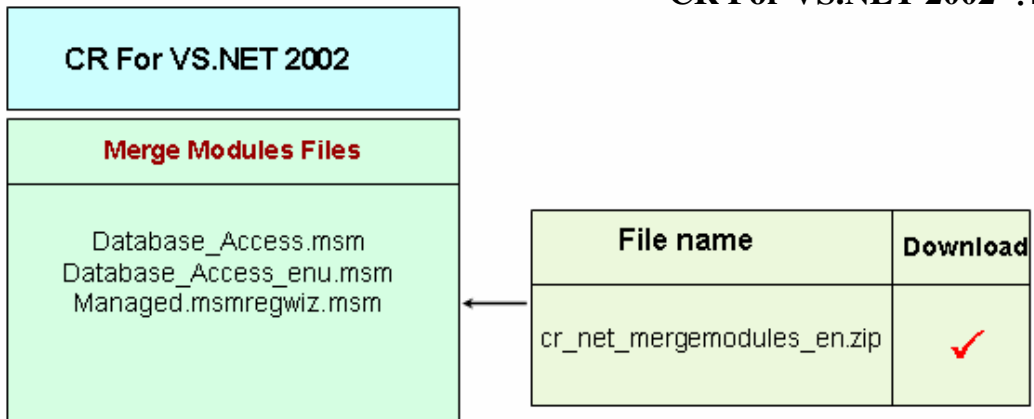
زمانی که شما می خواهید برای برنامه خود **Setup** تهیه نمایید چنانچه از **CR** استفاده کرده باشید باید از **Merge Module** های نسخه استفاده شده **CR** استفاده کنیم. حال مروری خواهیم داشت بر **Merge Module** های نسخه های مختلف **CR**. توجه داشته باشید که بعضی از **Merge Module** به صورت اتوماتیک نصب می شود و بعضی دیگر را باید از سایت خود **Business Object** دریافت نمایید.

1. Crystal Reports 9.0**2. Crystal Reports 10.0**

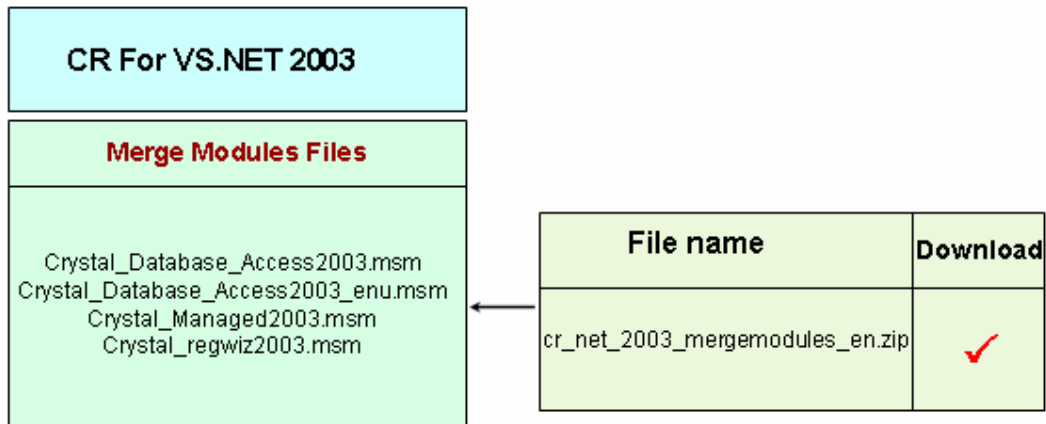
3. Crystal Reports 11.0



4. CR For VS.NET 2002



5. CR For VS.NET 2003



برای کسب اطلاعات بیشتر به آدرس زیر مراجعه نمایید:

http://support.businessobjects.com/downloads/merge_modules.asp

مروری بر Crystal Reports Developer Edition

مدتی بود که (فکر کنم الان هست) بحث طراحی گزارشات دلخواه یکی از مباحث داغ برنامه های گزارش سازی بود ولی باید گفت که شما نباید برای هر برنامه ای که نوشتید این امکان را برای کاربران فراهم سازید زیرا :

1. آموزش آن به کاربران آسان نیست و از لحاظ زمان بسیار پرهزینه است.
2. هر برنامه مستحق گزارشگیری دلخواه نیست (فقط نرم افزارهای بزرگ مستحق این امکان هستند).
3. فارسی سازی آن وقتگیر است.
4. قیمت پروژه را بسیار بالا می برد.
5. و ...

پس شما باید تمامی جوانب کار را در نظر بگیرید.

در بازار نسخه Crystal Reports XI Developer Edition R2 موجود می باشد و شما می توانید این نسخه که آخرین نسخه آن نیز می باشد تهیه نمایید.

نسخه معرفی شده برای NET. نیز مورد استفاده قرار می گیرد و Merge Module های آن برای NET. عبارتند از :

CrystalReports11_maps.msm
CrystalReports11_RDC_Designtime.msm
CrystalReports11_RDC_License.msm
CrystalReports11_RDC_Reportengine.msm
CrystalReports11_RDC_Runtime.msm

امیدوارم که نکات بالا فراموش نشود.

نوشته شده توسط فرید طاهری

با تشکر از دوستان گرامی آقایان احمد غفاری – مسعود طاهری که مرا در تهیه این Doc همراهی نمودند امید است این Doc به اندوخته های علمی شما دوستان چیزی اضافه نموده باشد گرچه اندک.

کمی در مورد In_Chan_Nafar :

این گروه تشکیل شده است از

- ❖ مهندس مسعود طاهری
- ❖ فرید طاهری (دانشجو)
- ❖ احمد غفاری (دانشجو)
- ❖ مجید طاهری (دانشجو)
- ❖ خدیجه زیبا.. پور (دانشجو)
- ❖ مهندس محسن جعفری

مفداری در مورد سبک ها

متدولوژی آنالیز : RUP

زبان یکپارچه سازی : UML

سبک برنامه نویسی : Multi tier – XP

زبان برنامه نویسی : C# , VB.NET

پایگاه داده : SQL Server 2005 و...

گزارش : Crystal Reports

گزارش سازی دلخواه : در صورت درخواست

و...

اگر تونستم قسمت دوم این آموزش رو هم واستون می نویسم.

موفق باشید

نوروز 1386